EVALUATING THE PERFORMANCE OF MULTIPLE CLASSIFIER SYSTEMS: A MATRIX ALGEBRA REPRESENTATION OF BOOLEAN FUSION RULES

THESIS

Justin Mitchell Hill, Captain, USAF

AFIT/GOR/ENC/03M-01

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Research sponsored in part by the Air Force Research Laboratory, Air Force Materiel Command, USAF. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this thesis are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

AFIT/GOR/ENC/03M-01

EVALUATING THE PERFORMANCE OF MULTIPLE CLASSIFIER SYSTEMS: A MATRIX ALGEBRA REPRESENTATION OF BOOLEAN FUSION RULES

THESIS

Presented to the Faculty of the Graduate School of Engineering and Management

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Justin Mitchell Hill, B.S.

Captain, USAF

March 2003

AFIT/GOR/ENC/03M-01

EVALUATING THE PERFORMANCE OF MULTIPLE CLASSIFIER SYSTEMS: A MATRIX ALGEBRA REPRESENTATION OF BOOLEAN FUSION RULES

THESIS

Justin Mitchell Hill, B.S.

Captain, USAF

Approved:

Mark E. Oxley, Ph.D. Advisor	Date
Kenneth W. Bauer, Jr., Ph.D. Reader	Date

Acknowledgements

None of this would have been possible without much appreciated guidance from my committee, and I would likely not have maintained my sanity without the continued love and support of my wife.

Justin Mitchell Hill

Table of Contents

Page

Acknowledgements iii

List of Figures vi

List of Tables vii

Abstract viii

I. Introduction 1
  1.1 General Discussion and Background 1
  1.2 Problem Description 1
  1.3 Research Objectives 3

II. Literature Review 4
  2.1 Introduction/Overview 4
  2.2 Receiver Operating Characteristic (ROC) Curves 4
    2.2.1 ROC Curve Background 4
    2.2.2 Comparing ROC Curves 6
  2.3 Fusing Classifiers 7
    2.3.1 Categories of Decision Combination 8
    2.3.2 Independence 9
  2.4 Different Methods for Fusing Classifiers 10
    2.4.1 Constant False Alarm Rate 10
    2.4.2 Boolean Fusion Rules 11
    2.4.3 Identification System Operating Characteristic Curves 14
    2.4.4 ROC Fusion 16

III. Research Methodology and Derivation 18
  3.1 Introduction 18
  3.2 Notation 18
  3.3 Classifier Performance 19
    3.3.1 Conditional Performance Matrix 19
    3.3.2 Prior Probabilities Matrix 20
    3.3.3 Joint Performance Matrix 22
  3.4 *Across* Fusion State Probabilities 23
    3.4.1 Joint State Probabilities Matrix 23
    3.4.2 Combined Prior Probabilities Matrix 27
    3.4.3 Conditional State Probabilities Matrix 28
    3.4.4 Truth Matrix 29
    3.4.5 Fusion Rule Matrix 31
    3.4.6 System Joint Performance Matrix 35

Page

3.4.7 System Prior Probabilities Matrix 36
3.4.8 System Conditional Performance Matrix 37
3.4.9 Summary 38
3.5 *Within* Fusion State Probabilities 38
3.5.1 Conditional State Probabilities Matrix 39
3.5.2 Combined Prior Probabilities Matrix 41
3.5.3 Joint State Probabilities Matrix 42
3.5.4 Truth and Fusion Rule Matrices 43
3.5.5 Summary 43
3.6 Estimating ROC Curves 44
3.6.1 Overview 44
3.6.2 ROC Fusion 44
3.6.3 Lagrangian Optimization 45
3.6.4 Brute Force 45
3.7 Summary 46

IV. Summary and Recommendations 47
4.1 Overview 47
4.2 Summary of Contributions 47
4.3 Recommendations for Future Research 48

Bibliography 49

List of Figures

Figure		Page
1.	Notional Single Classifier System.	2
2.	Notional Multiple Classifier System with Two Classifiers.	3
3.	Typical Receiver Operating Characteristics (ROC) Curve.	5
4.	Event, Feature and Label Sets for a Single Classifier System.	18
5.	Classification Errors for a Single Classifier System	23
6.	Event, Feature and Label Sets for a Two-Classifier System Combining Decisions *Across* Target Types	24
7.	Classification Errors for a Two-Classifier System in Which Decisions are Combined Using the Logical AND Rule.	32
8.	Classification Errors for a Two-Classifier System in Which Decisions are Combined Using the Logical OR Rule.	33
9.	Event, Feature and Label Sets for a Multiple Classifier System Combining Decisions *Within* Target Types.	39

List of Tables

Table		Page
1.	Conditional Classification Probabilities.	5
2.	Conditions for Statistical Independence.	9
3.	Relevant Fusion Rules for a 3-Classifier MCS.	12
4.	Categories of Relevant Boolean Fusion Rules for a 3-Classifier System	13
5.	Ralston's Performance Matrix.	14
6.	Conditional Performance Matrix for a Classifier with Two Labels.	19
7.	Joint State Probabilities Matrix Composed from 2×2 JPMs.	26
8.	Conditional State Probabilities Matrix Composed from 2×2 CPMs.	29
9.	Result After Post-Multiplying a JSPM by a Truth Matrix.	30

AFIT/GOR/ENC/03M-01

Abstract

Given a finite collection of classifiers one might wish to combine, or fuse, the classifiers in hopes that the multiple classifier system (MCS) will perform better than the individuals. One method of fusing classifiers is to combine their final decision using Boolean rules (e.g., a logical OR, AND, or a majority vote of the classifiers in the system). An established method for evaluating a classifier is measuring some aspect of its Receiver Operating Characteristic (ROC) curve, which graphs the trade-off between the conditional probabilities of detection and false alarm. This work presents a unique method of estimating the performance of an MCS in which Boolean rules are used to combine individual decisions. The method requires performance data similar to the data available in the ROC curves for each of the individual classifiers, and the method can be used to estimate the ROC curve for the entire system. A consequence of this result is that one can save time and money by effectively evaluating the performance of an MCS without performing experiments.

EVALUATING THE PERFORMANCE OF MULTIPLE CLASSIFIER SYSTEMS: A MATRIX ALGEBRA REPRESENTATION OF BOOLEAN FUSION RULES

I. Introduction

1.1 General Discussion and Background

The ability to accurately detect and identify targets is an important issue for the U. S. Air Force and the Department of Defense. The military departments traditionally wish to determine whether a particular object is hostile or friendly (target or clutter, foe or friend, etc.). Similar problems exist in many fields. For example, members of the medical community may wish to distinguish between cancerous and benign cells, or a mortgage company might attempt to discern a fit borrower from one who is likely to default on a loan.

All these problems are a part of a broad field called classification, a field which includes several approaches for solving problems like these. Methods could include something as simple as a visual inspection of the object of interest or a more scientific approach, like the linear discriminant function developed by Fisher [7]. Another common approach to modern classification problems is the use of artificial neural networks, which use algorithms that *learn* how to classify data. In the early 1980s, scientists and engineers began examining the idea of using groups of classifiers, or multiple classifier systems (MCSs), in hopes of increasing accuracy, and research in this area of the field continues today [23].

1.2 Problem Description

Researchers are often concerned with evaluating the performance of a classifier or an MCS. The systems employed in these applications are usually systems using sensors to collect data and other mechanisms, called classifiers, to classify each observation. Figure 1 shows a notional single

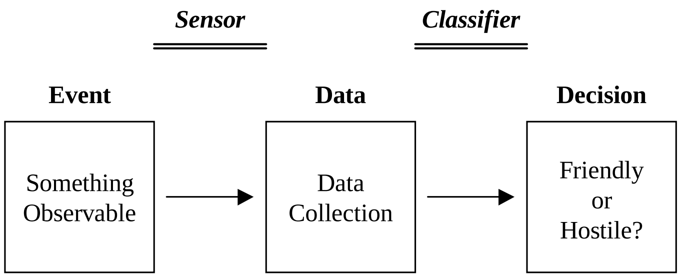

Figure 1: Notional Single Classifier System.

classifier system. The data collected by the sensor, called features, are typically converted by the classifier to some numerical value. If this value is greater than a nominal threshold value, the observation is placed into one class. Otherwise the observation is placed into the other class.

A multiple classifier system is one in which multiple sensor/classifier ensembles collect data and reach conclusions independently. Those decisions are then combined in some manner to reach a decision for the system. A system designer might want to build a system that makes use of multiple classifiers for various reasons. Different classifiers may be trained to detect:

- Different types of targets.
- Different attributes of the same target.
- The same target under different operating conditions.

In a multiple classifier system there are potentially several events observed and several streams of data collected. Figure 2 depicts the design of a notional two-classifier system in which the final classifier decisions are combined. Combining the final decision of each classifier in the system is not the only method of combination in an MCS, but discussion in Chapter 2 explains why researching this type of classifier combination is a worthwhile pursuit, especially for military applications.

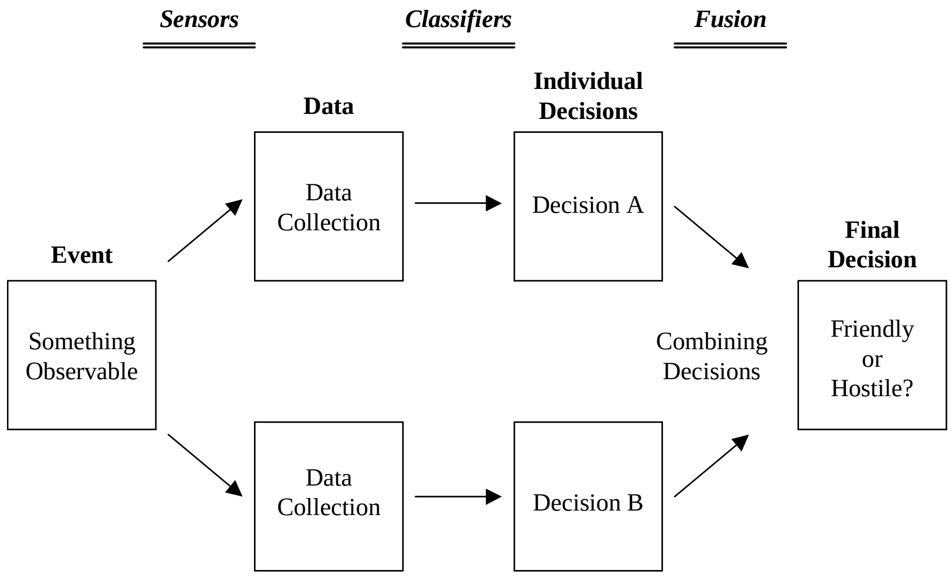

Figure 2: Notional Multiple Classifier System with Two Classifiers.

1.3 Research Objectives

The goal of this research is to provide some mathematical insight into the process of evaluating MCSs. Specifically, the purpose of the research is to identify properties that may aid the analyst or engineer in evaluating an MCS by using the data available in the Receiver Operating Characteristic (ROC) curves from the individual classifiers in the ensemble. The primary contribution of the thesis is a formula for computing ROC curve values for an MCS using (a) the ROC curves from the individual classifiers in the system, and (b) any Boolean rule for combining the decisions from the individuals.

II. Literature Review

2.1 Introduction/Overview

This chapter reviews the literature regarding two-class classification or detection. Receiver operating characteristic (ROC) curves are described as a visual method for evaluating classifier performance, and methods for comparing classifiers using ROC curves are discussed. Then the concepts of combining and fusing classifiers are introduced, followed by a discussion of various systems of multiple classifiers and the different methods used to bring together results from the individual classifiers to optimize the performance of the system. This discussion will include an introduction to the concept of constant false alarm rate (CFAR) fusion, where the system is designed to yield the best possible detection rate while maintaining an acceptable number of false alarms. There will also be considerable discussion of simple logical, or Boolean, rules for combining classifier outputs.

2.2 Receiver Operating Characteristic (ROC) Curves

2.2.1 ROC Curve Background. Receiver operating characteristic (ROC) curves are one way of describing the performance of a classifier. Once the classifier makes a decision (friendly or hostile), there are four possible results, or output states. The classifier can:

1. Correctly identify a hostile target.
2. Misclassify a hostile target as friendly.
3. Misclassify a friendly target as hostile.
4. Correctly identify a friendly object or clutter.

Using Egan's terminology [6], the probability of scenario one (conditional upon the existence of a hostile target) corresponds with the *hit rate* or *probability of detection* and will be denoted P_D. Scenario two corresponds with the *miss rate* and will be represented by P_M. Scenario three

Table 1: Conditional Classification Probabilities.

Scenario	Notation	Meaning
1. Hit Rate	P_D	Pr{Hostile Classification | Hostile Target Present}
2. Miss Rate	P_M	Pr{Friendly Classification | Hostile Target Present}
3. False Alarm Rate	P_{FA}	Pr{Hostile Classification | No Hostile Targets Present}
4. Correct Rejection Rate	P_C	Pr{Friendly Classification | No Hostile Targets Present}

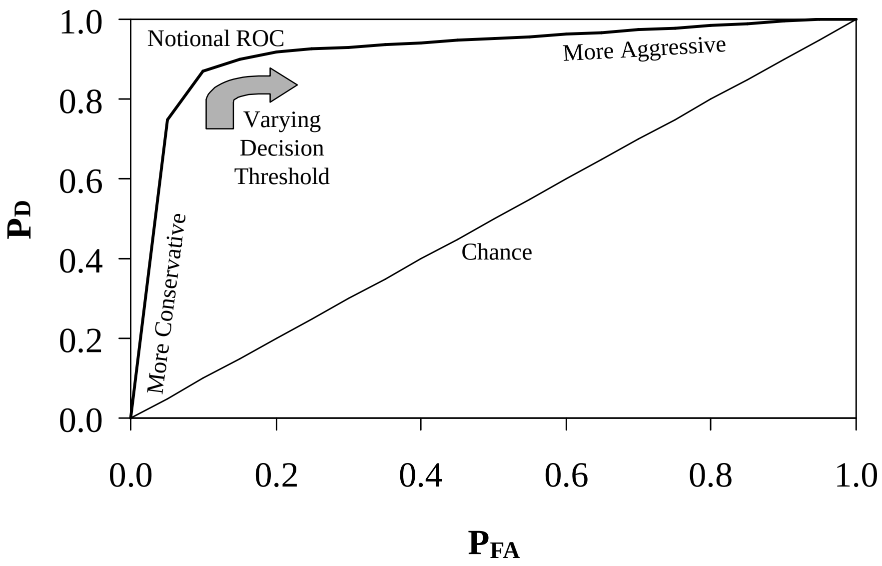

Figure 3: Typical Receiver Operating Characteristics (ROC) Curve.

corresponds with the *false alarm rate* and will be denoted P_{FA}. Finally, the term P_C is used to signify scenario four, a correct rejection. Table 1 lists the conditional probabilities for a classifier. Note that $P_D + P_M = 1$ and $P_{FA} + P_C = 1$.

The ROC curve is a graph of the trade-off between the target detection rate (P_D) and the false alarm rate (P_{FA}) for a particular classifier. A notional ROC curve is shown in Figure 3. Typically, the ROC curve is constructed by varying the decision threshold (θ) for the classifier and plotting the observed values for P_D and P_{FA}; therefore, a ROC curve for a typical classifier is actually a two-dimensional projection of an object in 3-space (θ, P_D, P_{FA}) [2].

Egan notes that the slope at any point on a ROC curve is equal to the likelihood ratio for a particular observation x. The likelihood ratio is computed with the function

$$L(x) = \frac{\Pr\{x \mid H\}}{\Pr\{x \mid F\}},$$

and is a measure of the strength of the evidence carried by the observation. He points out that a decision rule can be based on this measure (e.g., if $L(x) > \theta_{L(x)}$, conclude hostile). That is, if the evidence is stronger than some predetermined value $\theta_{L(x)}$, the classifier will determine that the observation is hostile. Thus, higher values of $\theta_{L(x)}$ correspond with more conservative decisions, because the classifier requires more evidence. Decreasing $\theta_{L(x)}$ yields a more aggressive fusion rule.

The alternative is for the classifier to base its decision on the actual observation x. If $L(x)$ is monotonic with respect to x these decision rules are similar (e.g., if $L(x) > \theta_{L(x)}$ or $x > \theta_x$, conclude that the object is hostile). However, if $L(x)$ is not monotonic with respect to x one can still define a decision rule based on x that is equivalent to the rule based on $L(x)$. In such cases the classifier would conclude that the observation is hostile if the value x was included in specified subintervals. (Egan provides an example of such a case which, for brevity, is omitted here.) Constructing a ROC curve by varying θ_x across x in a case like this may result in an ill-behaved ROC curve.

If the decision rule is based on $L(x)$ rather than x, the slope of the ROC curve must be nonincreasing with respect to P_{FA} since the decisions are always more aggressive as $\theta_{L(x)}$ decreases. Egan refers to a ROC curve constructed in this manner as a *proper ROC curve*, and he notes that the slope of a proper ROC curve is nonincreasing with respect to P_{FA}. That is, a proper ROC curve is always concave down. Logically, if the evidence that the object is hostile is strong, $P_D >> P_{FA}$, and as the strength of the evidence decreases P_D will approach P_{FA}.

2.2.2 Comparing ROC Curves. There are several methods for comparing ROC curves. The most obvious comparison is visual. If one curve always has a higher P_D than another for a

given P_{FA}, then the classifier corresponding to the higher curve is superior. These cases are not very interesting, however, and other methods of comparison exist for evaluation in cases where one curve is not always superior. One commonly used method is to compare the area under the curve (AUC) [6], [12], [3]. Classifiers that correspond with curves having greater AUCs are considered better. The AUC can be computed using a trapezoidal approximation of the area. Another method is to compute the average metric distance between the ROC curve and the chance line [1]. This distance can be computed by

$$MD = \frac{\sum_{i=1}^{n} \left\| \left(P_D\left(\theta_i\right), P_{FA}\left(\theta_i\right)\right) - \left(\theta_i, \theta_i\right) \right\|_1}{n},$$

where $\left(P_D\left(\theta_i\right), P_{FA}\left(\theta_i\right)\right)$ denotes the i^{th} point sampled from the ROC curve and $\left\|\cdot\right\|_1$ is the 1-norm on $\mathbb{R}^2$. The classifier with the larger MD is considered superior. A detailed discussion of ROC curve comparison methods (including the derivation of the average metric distance and a multinomial selection algorithm) is found in Alsing [1].

2.3 Fusing Classifiers

Saranli and Demirekler note that "decision combination systems are of considerable interest to a large number of pattern recognition fields" [25]. They also noted some potential benefits that may be gained by combining classifiers. Bayesian classifiers work by estimating the posterior probability of a particular observation belonging to a particular class, and in statistical estimation there is inherent variance in the estimate because it is based on sample data. Tumer and Ghosh showed that averaging the estimates from several different classifiers reduces this variance [26]. Saranli and Demirekler also pointed out the rather obvious notion that some classifiers may work better than others for certain observations or particular types of problems [25]. By using several *experts* the system designer hopes to have a better chance at correct classification.

2.3.1 Categories of Decision Combination. Dasarathy notes that classifier decision fusion is a subset of sensor fusion. He defines sensor fusion as "the study of optimal information processing in distributed multisensor environments through intelligent integration of the multisensor data" [5], and he notes that there is a three-level hierarchy of fusion, comprised of *data*, *feature*, and *decision* fusion. *Data* is defined as raw information which can be organized or combined to create *features* relating to a particular observation. The features can then be used by a classifier to arrive at a *decision*. Dasarathy defines five separate categories of fusion problems based on input and output modes.

1. Data input/Data output
2. Data input/Feature output
3. Feature input/Feature output
4. Feature input/Decision output
5. Decision input/Decision output

The final category, where individual decisions are used as input to arrive at a system decision, is an especially important category and the focus of this research. Dasarathy notes that this type of fusion is applicable no matter what types classifier systems are employed. By using only the final decision from each classifier, the system is not hindered by instances of incompatibility. The individuals may be designed using different architectures, methodologies, or philosophies; but a system combining the discrete decision values is not affected by such disparities in design. Further, Varshney notes that many problems have practical limitations on the amount of data that can be transferred from the individual classifiers to the fusion center [28]. In these problems, the geographical proximity of the individual classifiers and the bandwidth available for electronic communication contribute to a situation in which it may be beneficial to transmit as little data as possible.

Table 2: Conditions for Statistical Independence.

Conditions for Statistical Independence
$\Pr\{A \mid B\} = \Pr\{A\}$
$\Pr\{B \mid A\} = \Pr\{B\}$
$\Pr\{A \cap B\} = \Pr\{A\}\Pr\{B\}$

The paper by Xu *et al* is a seminal work in the field [30]. They further categorized the types of decision combination into three groups [30]. In Type I, only the final decision made by each classifier is sent to the fusion center (e.g., Class = B). In Type II each classifier reports a ranked list of possibilities, and the combiner uses the ranked list to make a decision (e.g., 1st Choice = B, 2nd Choice = A, etc.). A Type III combiner accepts a list of possible decisions along with some measure of confidence in those decisions (e.g., A = 0.60, B = 0.25, C = 0.15). The focus of their work was on Type I decision combination "due to its generality".

2.3.2 Independence. Analysis of multiple classifier systems often includes some discussion on *independence*. However, the research often seems conflicted about what type of independence is important. Some of the literature discusses statistical independence between the classifiers [21] [14], while other works are more concerned with the classifiers making independent errors [13] [24]. Others point out that combining classifiers that make negatively correlated errors can enhance system performance [17].

Statistical independence between two events A and B is defined if any of the conditions in Table 2 are true [29]. The primary reason for an assumption of statistical independence is that it makes computing joint probabilities much simpler. With this assumption, analysts can easily examine MCSs by computing joint probabilities without accounting for correlation between the individual classifiers. It may also make sense to ignore correlation in a notional analysis unless a realistic estimate of dependence is available. That said, an in-depth analysis of a specific MCS should probably include an analysis and discussion of classifier dependence. That is, *general analyses should* probably assume statistical independence, and *specific analyses should not.*

Hansen and Salamon assumed that the individual classifiers in a theoretical MCS made independent random errors [13], and many other researchers followed suit. Furthermore, Giacinto *et al*, observe that "most combination methods described in the literature assume that MCSs are made up of classifiers making independent classification errors" [10]. However, Kuncheva and her colleagues point out that negatively correlated errors can enhance MCS performance [17], so perhaps it is not independent errors, but opposing errors that are a key element in MCS construction.

2.4 Different Methods for Fusing Classifiers

This section presents a brief overview of some of the philosophies for fusing decisions. These methodologies were selected based on their foundations in optimization theory or their applicability to the remainder of the thesis.

2.4.1 Constant False Alarm Rate. When the objective of the classifier system is to detect a target in clutter it is conceivable that the clutter patterns might vary from target to target. Furthermore, if a target is mobile the clutter patterns around that particular target may not be stationary. One philosophy for detecting targets in scenarios like these is called Constant False Alarm Rate (CFAR) [28]. In CFAR fusion, the decision thresholds used by the classifiers in the MCS are allowed to vary in such a way that the detection probability is maximized while the system maintains a specified false alarm rate. Two popular approaches for CFAR are Cell Averaging CFAR (CA-CFAR) and Order Statistics CFAR (OS-CFAR). A typical method for CFAR optimization is the use of the Lagrangian expression given by

$$L(\theta_1, \theta_2, ..., \theta_k) = P_D(\theta_1, \theta_2, ..., \theta_k) - \lambda(P_{FA}(\theta_1, \theta_2, ..., \theta_k) - p),$$

where p is the maximum allowable false alarm rate for the system and the θ_i are the decision thresholds for the individual classifiers. Setting the gradient of L equal to the zero vector and solving for λ and the θ_i yields candidates for the optimal threshold values.

2.4.2 Boolean Fusion Rules. One method for combining classifier decisions is to use Boolean, or *simple,* rules. One simple rule is for the system to conclude that a hostile target is present if and only if all the classifiers in the system conclude that their observations both indicate hostile targets. This rule is heretofore referred to as the AND rule. Another simple rule is for the system to conclude that a hostile target is present if any of the individual classifiers indicate hostile targets. This rule is called the OR rule. When the system includes more than two classifiers, other simple rules are possible. For example, in a system where each classifier can only output two labels (f or h), the number of possible Boolean rules is 2^{2^K}, where K is the number of individual classifiers.

Liggins gives a subset of the possible Boolean rules for a three classifier system in a table similar to Table 3 [18]. The first three columns of the table indicate the decisions of the individual classifiers (1 for hostile, 0 for friendly). The remaining columns are vector representations of the fusion rules. A "1" in a particular row indicates that the system will conclude *hostile* if the outputs of the individual classifiers correspond with those in the first three columns. For example, under rule r2 the system will only classify an object as hostile if all three classifiers conclude the same, and under r4 the system ignores the decision of classifier A_3 and classifies an object as hostile if $A_1$and A_2 decide *hostile.*

He notes that the 256 theoretical rules for the 3-detector case can be reduced to 18 rules by application of the monotonicity rule, which assumes that P_D must be greater than P_{FA}. The 18 relevant rules represent "all possible physical contingencies." For example, a rule represented by $(0, 1, 0, 0, 0, 0, 0, 0)^T$ is not a relevant rule. Under such a rule the system would conclude *hostile* only if A_3 concluded *hostile* **and** the other classifiers concluded *friendly.* This rule would be illogical,

Table 3: Relevant Fusion Rules for a 3-Classifier MCS.

Classifiers			**Monotonic Fusion Rules**									
A_1	A_2	A_3	r1	r2	r4	r6	r8	r16	r18	r20	r22	r24
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	1	1	0	0	1	1
1	1	0	0	0	1	0	1	1	0	1	0	1
1	1	1	0	1	1	1	1	1	1	1	1	1
A_1	A_2	A_3	r34	r52	r56	r64	r86	r88	r96	r120	r128	r256
0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	1	1	1	1	1
0	1	0	0	1	1	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

because it ignores other cases where A_3 decides that the target is hostile. Thus, any relevant fusion rule represented with a 1 in the second row will also have a 1 in the fourth, sixth and eighth rows. Also note that rules r1 and r256 are not relevant fusion rules. They were included merely to bound the set of relevant rules.

The rules in Table 3 can be grouped into three classes: 1-classifier rules, 2-classifier rules and 3-classifier rules. Liggins notes that rules r16, r52 and r86 make up the subset of 1-classifier rules (e.g., for r16 the system concludes hostile if classifier A_1 concludes hostile). He also notes that the 2-classifier rules can be broken into AND rules and OR rules. From this point forward, AND rules may be indicated with a $\wedge$, and OR rules may be represented with $\vee$. Liggins also categorizes AND and OR rules for the 3-classifier case, as well as the majority vote and a case Liggins defines as *sensor dominance*, where the system always accepts the decision of one classifier but accepts the decisions of the other two classifiers only if they both agree that a hostile target is present. He fails to categorize the remaining three rules (r8, r20 and r22), all of which are similar. In these rules one sensor must conclude hostile and at least one of the others must agree. Such a configuration

Table 4: Categories of Relevant Boolean Fusion Rules for a 3-Classifier System

Category	**Rule**	**Meaning**
1-Classifier Rules		
	r16	A_1
	r52	A_2
	r86	A_3
2-Classifier Rules		
AND Rules	r4	$A_1 \wedge A_2$
	r6	$A_1 \wedge A_3$
	r18	$A_2 \wedge A_3$
OR Rules	r64	$A_1 \vee A_2$
	r96	$A_1 \vee A_3$
	r120	$A_2 \vee A_3$
3-Classifier Rules		
AND Rule	r2	$A_1 \wedge A_2 \wedge A_3$
OR Rule	r128	$A_1 \vee A_2 \vee A_3$
Majority Vote	r24	$(A_1 \wedge A_2) \vee (A_1 \wedge A_3) \vee (A_2 \wedge A_3)$
Sensor Dominance	r34	$A_1 \vee (A_2 \wedge A_3)$
	r56	$A_2 \vee (A_1 \wedge A_3)$
	r88	$A_3 \vee (A_1 \wedge A_2)$
Sensor Corroboration	r8	$A_1 \wedge (A_2 \vee A_3)$
	r20	$A_2 \wedge (A_1 \vee A_3)$
	r22	$A_3 \wedge (A_1 \vee A_2)$

will henceforth be called *Sensor Corroboration*, because at least one sensor must corroborate the decision of the primary sensor. A practical example of case where sensor corroboration would be appropriate is a threat detection scenario in which one classifier is trained to detect movement and the other two are each trained to identify a certain type of enemy vehicle. If the first classifier detects movement and one of the others confirms that the object is an enemy vehicle, then the object should be classified as a viable threat. Table 4 categorizes the 18 relevant rules.

A majority vote among the available classifiers is a simple rule that has received much attention in the literature [13] [30] [8] [15] [19] [9] [16] [?]. Voting rules are among the easiest to conceptualize, because everyday decisions are often made in this manner. Xu *et al* noted that all voting rules are not necessarily majority votes. One can specify a more conservative rule (e.g., require a 2/3 majority) or a less conservative rule such as a multi-class problem where the system decision is the class with the largest number of votes, whether or not there is a majority.

Table 5: Ralston's Performance Matrix.

Output State	**Truth**	
Classifier k	Friend	Hostile
1	$\Pr\{1 \mid F\}$	$\Pr\{1 \mid H\}$
.	.	.
.	.	.
.	.	.
m_k	$\Pr\{m_k \mid F\}$	$\Pr\{m_k \mid H\}$

Most theoretical analysis has been on the majority vote. Hansen and Salamon showed that the performance of the system will be greater than the performance of the best individual classifier under certain assumptions [13]. They noted that the classification accuracy of the system increases with the number of classifiers under the assumptions that (a) each classifier is right at least half the time, (b) the classifiers make independent errors. Matan gives both upper and lower bounds on classification accuracy for the more general case of a "k-of-n special majority", where n is the number of classifiers in the MCS and k is any integer greater than $n/2$ [19]. Kuncheva et. al also derive upper and lower bounds for the majority vote, but their work also takes into account the pairwise dependence between the classifiers in the MCS [16]. None of the literature discusses theoretical limits on the ROC curve of a majority voting system or any other MCS for that matter.

2.4.3 Identification System Operating Characteristic Curves. Ralston adapted the concepts of likelihood ratios to determine the best choice of combined classifier output states [22]. Given K classifiers, each with m_k output states, the purpose of Ralston's combat identification system is to determine whether an exemplar is a *friend* or *hostile*. Each classifier $k \in \{1, 2, ..., K\}$ has a *performance matrix* (denoted PM and determined via system testing, etc.) with two columns. The first column corresponds to the probability of a particular output state given that the exemplar is truly a friend. The second column gives the probability of each output state given that the exemplar is hostile. Table 5 shows an example PM.

There is a system output state corresponding with each possible combination of outputs from the individual classifiers in the system. Thus, if there are n_k output states for each classifier k,

there are

$$N = \prod_{k=1}^{K} n_k$$

output states for the system. Assuming that the classifiers in the system are not correlated, the conditional probability that a friendly object will yield any output state can be computed as the product of the appropriate cells from the first column of the PM. For example, one output state for a notional system occurs when classifier A_1 is in state q, A_2 is in state r, and A_3 is in state s. The probability that a friendly object puts the system in this state is the product of the $(q, 1)$ cell of the PM for A_1, the $(r, 1)$ cell of the of the PM for A_2, and the $(s, 1)$ cell of the PM for A_3. The probability that a hostile object will yield a particular output state can be computed in a similar manner.

Also worth mentioning is Ralston's representation of possible output state combinations. He suggests representing each rule or output state combination as a vector R of length N. If the rule indicates that a particular output state j will force the system to conclude that an object is hostile, the vector R contains a 1 in the j^{th} element of R. Otherwise, R contains a 0. With this representation of the fusion rule, Ralston was able to determine P_D and P_{FA} with the following formulas:

$$P_D = \sum_{i=1}^{N} \Pr\{j \mid H\} \cdot R(j)$$
$$P_{FA} = \sum_{i=1}^{N} \Pr\{j \mid F\} \cdot R(j)$$

Ralston then defines an Identification System Operating Characteristic (ISOC) curve by computing the likelihood ratio $\Pr\{j \mid H\} / \Pr\{j \mid F\}$ for each system output state j ordering the likelihood ratios from greatest to least. The output state with the highest likelihood ratio is the most conservative output state and will produce the best possible P_D for its P_{FA}. The next output state is a combination of the previous state with the next most conservative state, and so on. By plotting

the P_D and P_{FA} values for these successive combinations of output states Ralston is able to provide the optimal combination of states for each P_{FA} without enumerating all 2^N possible output state combinations. This is analogous to a ROC curve for a single classifier whose decision is based on $L(x)$. Instead, Ralston treats x as a single output state instead of a single observation [6].

2.4.4 ROC Fusion. Oxley and Bauer presented a novel approach for classifier system evaluation by showing that it is possible to analytically construct the ROC curve for an MCS based on certain fusion rules (AND and OR) using only data from the ROC curves for the individual classifiers in the system [20]. The purpose of the classifier systems researched in their work was to determine if the system was in one of two states, (e.g., *friendly* or *hostile)*. Their work resulted in four primary contributions.

First they defined the difference between fusion *within* and *across* target types. A system of classifiers that are fused *within* is a system in which all classifiers are trained to detect a particular type of target. Thus, they share the same prior probability of detection. Moreover, there are only two possibilities for *truth* in such a system. Either the target is present, or it is not. A system that is fused *across* target types includes classifiers trained to detect a number of target types. Each of these types of targets may have a different prior probability of detection, and since the system seeks different types of targets, it can *accidentally* arrive at the correct conclusion if a classifier seeking target type A incorrectly detects a target when a target type B is present. For reasons such as these, an *across* system may be more difficult to analyze than a *within* system.

The second contribution was the derivation of formulas for P_D and P_{FA} for logical AND and OR rules in *within* and *across* systems. The third and fourth contributions were very closely related. Rather than a traditional definition for a ROC curve (P_D vs. P_{FA}), Oxley and Bauer defined a ROC curve as the maximum value of P_D for each possible P_{FA} for that particular classifier. Although this contribution may seem trivial, it allowed Oxley and Bauer to analytically determine the ROC curves for logical AND and OR rules.

The example used was a system designed to solve a two-class problem in which there were (a) two-classifiers, (b) each classifier could output two labels, and (c) the system could output two labels. However, Chapter 3 of this document shows that the results for AND and OR can be extended to any number of classifiers and labels.

III. Research Methodology and Derivation

3.1 Introduction

This chapter provides a matrix algebra representation for evaluating the performance of Boolean fusion rules in an MCS designed for two-class classification. The representation is general in that it accommodates rules for fusing *within* and *across* target types and that it allows for any number of classifiers, each of which can output any number of labels.

Assume there is a classifier trained to detect a hostile target. Formally, consider a set of events $\mathcal{E}$, which can be divided into two subsets. One subset consists of instances of a hostile target ($\mathcal{E}_h \subset \mathcal{E}$). The other subset ($\mathcal{E}_f \subset \mathcal{E}$) consists of objects not belonging to $\mathcal{E}_h$ and corresponds to friendly objects. A sensor S maps an event to a feature set $\mathcal{X}$. Thus, each feature vector x is a random vector since is is the image of the random variable S. Events in the subset $\mathcal{E}_h$ map to a subset of $\mathcal{X}$ called $\mathcal{X}_h$, and events in $\mathcal{E}_f$ map to $\mathcal{X}_f \subset \mathcal{X}$. Let $\times$ be a threshold set (or a set of parameters) used by $A(x, \theta \in \times)$ to map each feature vector to a label set $\mathcal{L} = \{f, h\}$. That is, $l = A(x, \theta)$ and $A(x, \theta) : \mathcal{X} \to \mathcal{L}$ for each $\theta \in \times$. Figure 4 illustrates this process.

3.2 Notation

Throughout the discussion assume there are a finite number, K, of classifiers, and each classifier is, in fact, a family of classifiers dependent upon a parameter, $\theta_k \in \Theta_k$. Each classifier A_k is coupled with a sensor S_k which maps events to the feature set $\mathcal{X}_k$, and outputs a label in

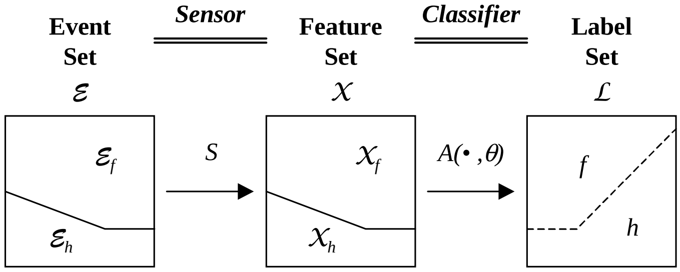

Figure 4: Event, Feature and Label Sets for a Single Classifier System.

Table 6: Conditional Performance Matrix for a Classifier with Two Labels.

		Feature Set (*Truth*)	
		$x_k \in \mathcal{X}_{k,f}$	$x_k \in \mathcal{X}_{k,h}$
	1	$\Pr\{A_k(x_k) = 1 \mid x_k \in \mathcal{X}_{k,f}\}$	$\Pr\{A_k(x_k) = 1 \mid x_k \in \mathcal{X}_{k,h}\}$
Output	2	$\Pr\{A_k(x_k) = 2 \mid x_k \in \mathcal{X}_{k,f}\}$	$\Pr\{A_k(x_k) = 2 \mid x_k \in \mathcal{X}_{k,h}\}$
Label	$\vdots$	$\vdots$	$\vdots$
	m_k	$\Pr\{A_k(x_k) = m_k \mid x_k \in \mathcal{X}_{k,f}\}$	$\Pr\{A_k(x_k) = m_k \mid x_k \in \mathcal{X}_{k,h}\}$

the label set $\mathcal{L}_k$. The cardinality of the label set $\mathcal{L}_k$ is m_k (i.e., there are m_k labels). A feature vector $x_k \in \mathcal{X}_k$ can indicate a friendly or hostile target, and there are two corresponding subsets of $\mathcal{X}_k$. $\mathcal{X}_{k,h}$ is the subset containing feature vectors that should indicate hostile targets, and $\mathcal{X}_{k,f}$ is the subset containing feature vectors that should indicate friendly targets. A classifier output denoting a hostile target is denoted by a lower case h ($A_k(x_k) = h_k$), and the output denoting a friendly target is denoted by a lower case f ($A_k(x_k) = f_k$). The event, feature, and label sets for the system will be denoted by a subscript S, not to be confused with the sensor S.

3.3 Classifier Performance

The following sections provide the reader with some tools for evaluating classifier performance. These tools are applicable to any classifier designed to solve a two-class problem; however, the examples presented are for a classifier that only outputs two labels (f and h).

3.3.1 Conditional Performance Matrix. One can summarize the performance of a classifier operating at a particular decision threshold θ in terms of the conditional probabilities in Table 1 by recording them in a matrix equivalent to the performance matrix defined by Ralston [22]. This matrix will be called the Conditional Performance Matrix (CPM). For each classifier $k \in \{1, 2, ..., K\}$, let C_k denote the CPM corresponding to the k^{th} classifier. Table 6 shows that each column corresponds with *truth* and each row corresponds with a particular output label.

To be consistent, the first row should correspond with the friendly label and the last row should correspond with the hostile label. Similarly, the first column should correspond with instances of

friendly objects, and the second column should correspond with instances of hostile targets. If this convention is kept, the last row of a 2×2 CPM identifies the P_{FA} and P_D for the classifier, and the set of CPMs for all $\theta \in \Theta_k$ can be used to construct the ROC curve for that classifier.

$$
\begin{aligned}
C_k &= \begin{bmatrix} \Pr\{A_k(x_k) = f_k | x_k \in \mathcal{X}_{k,f}\} & \Pr\{l_k = f_k | x_k \in \mathcal{X}_{k,h}\} \\ \Pr\{A_k(x_k) = h_k | x_k \in \mathcal{X}_{k,f}\} & \Pr\{l_k = h_k | x_k \in \mathcal{X}_{k,h}\} \end{bmatrix} \\
&= \begin{bmatrix} P_{C,k} & P_{M,k} \\ P_{FA,k} & P_{D,k} \end{bmatrix} \\
&= \begin{bmatrix} 1 - P_{FA,k} & 1 - P_{D,k} \\ P_{FA,k} & P_{D,k} \end{bmatrix}
\end{aligned}
$$

Definition III.1. *A* ***Conditional Performance Matrix (CPM)*** *for a classifier k is an $m_k \times 2$ matrix in which the columns correspond with truth, the rows correspond with the classifier's output labels, and the (i, j) cell is the conditional probability of the classifier outputting label i when the true state of the system is j. The sum of each column of the CPM is unity (i.e., the CPM is column stochastic).*

3.3.2 Prior Probabilities Matrix. Using the definition of conditional probability,

$$
\Pr\{A|B\} = \frac{\Pr\{A \cap B\}}{\Pr\{B\}},
$$

one can compute the joint probability $\Pr\{A \cap B\}$ by simply multiplying the conditional probability by the *a priori* probability $\Pr\{B\}$. Consequently, one can multiply each column of the CPM by the appropriate *a priori* probability to determine the unconditional probability of each output state. Since the two subsets $\mathcal{X}_{k,h}$ and $\mathcal{X}_{k,f}$ are complementary, the probabilities $\Pr\{x_k \in \mathcal{X}_{k,h}\}$ and $\Pr\{x_k \in \mathcal{X}_{k,f}\}$ are also complementary. Thus, one could multiply the first column of the CPM by $\Pr\{x_k \in \mathcal{X}_{k,f}\}$ and the second column by $\Pr\{x_k \in \mathcal{X}_{k,h}\}$ to determine the joint probabilities of

the output labels coinciding with a particular true state. The result is a matrix with the following construction.

$$\begin{bmatrix} \Pr\{A_k(x_k) = f_k \cap x_k \in F_k\} & \Pr\{A_k(x_k) = f_k \cap x_k \in H_k\} \\ \Pr\{A_k(x_k) = h_k \cap x_k \in F_k\} & \Pr\{A_k(x_k) = h_k \cap x_k \in H_k\} \end{bmatrix}$$

Let $\alpha_k = \Pr\{x_k \in H_k\}$, and $(1 - \alpha_k) = \Pr\{x_k \in F_k\}$. Define a 2×2 diagonal matrix ρ_k as follows.

$$\rho_k = \begin{bmatrix} (1 - \alpha_k) & 0 \\ 0 & \alpha_k \end{bmatrix}$$

The matrix ρ_k is called the Prior Probabilities Matrix for classifier k.

Definition III.2. *The* ***Prior Probabilities Matrix (PPM)*** *for a particular type of target is a* 2×2 *diagonal matrix in which the* $(2, 2)$ *cell is the probability of observing that type of target, and the* $(1, 1)$ *cell is a complementary value. Thus, the trace of a PPM is unity.*

3.3.3 Joint Performance Matrix. Now we can define the Joint Performance Matrix (JPM) for a two-label classifier as follows.

$$
\begin{aligned}
J_k &= \begin{bmatrix} \Pr\{\text{Friendly Label} \cap \text{ No Target}\} & \Pr\{\text{Friendly Label} \cap \text{ Target Present}\} \\ \Pr\{\text{Hostile Label} \cap \text{No Target}\} & \Pr\{\text{Hostile Label} \cap \text{ Target Present}\} \end{bmatrix} \\
&= \begin{bmatrix} \Pr\{A_k(x_k) = f_k \cap x_k \in F_k\} & \Pr\{A_k(x_k) = f_k \cap x_k \in H_k\} \\ \Pr\{A_k(x_k) = h_k \cap x_k \in F_k\} & \Pr\{A_k(x_k) = h_k \cap x_k \in H_k\} \end{bmatrix} \\
&= C_k \rho_k \\
&= \begin{bmatrix} \Pr\{A_k(x_k) = f_k | x_k \in \mathcal{X}_{k,f}\} & \Pr\{l_k = f_k | x_k \in \mathcal{X}_{k,h}\} \\ \Pr\{A_k(x_k) = h_k | x_k \in \mathcal{X}_{k,f}\} & \Pr\{l_k = h_k | x_k \in \mathcal{X}_{k,h}\} \end{bmatrix} \begin{bmatrix} \Pr\{x_k \in \mathcal{X}_{k,f}\} & 0 \\ 0 & \Pr\{x_k \in \mathcal{X}_{k,h}\} \end{bmatrix} \\
&= \begin{bmatrix} 1 - P_{FA,k} & 1 - P_{D,k} \\ P_{FA,k} & P_{D,k} \end{bmatrix} \begin{bmatrix} (1-\alpha_k) & 0 \\ 0 & \alpha_k \end{bmatrix} \\
&= \begin{bmatrix} (1-\alpha_k)(1 - P_{FA,k}) & \alpha_k (1 - P_{D,k}) \\ (1-\alpha_k) P_{FA,k} & \alpha_k P_{D,k} \end{bmatrix}
\end{aligned}
$$

The events associated with each element of the JPM are mutually exclusive and exhaustive, and the probabilities define the entire set of outcomes for the classifier k, or the probability of each region shown in the shaded portion of Figure 5. Note that the sum of the elements of J_k equals one, and the trace of a square JPM represents the classification accuracy for the classifier.

Definition III.3. *The* ***Joint Performance Matrix (JPM)*** *for a classifier k is a $m_k \times 2$ matrix in which the columns correspond with truth, the rows correspond with the classifier's output labels, and cell (i, j) is the probability of the classifier outputting label i when the system is in state j. The JPM gives the probabilities of all possible outcomes for the classifier. One can construct the JPM from the CPM and PPM using the formula $J_k = C_k \rho_k$, and the sum of the elements of the JPM is unity.*

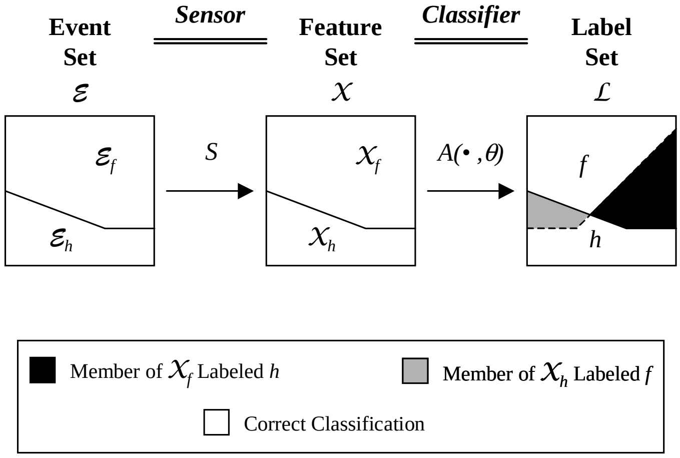

Figure 5: Classification Errors for a Single Classifier System

3.4 Across Fusion State Probabilities

Consider a system made up of K classifiers with similar constructions. Reconsider the set of events $\mathcal{E}$, which is now partitioned into k subsets: $\mathcal{E}_{h,k}$ consists of all hostile targets of type k (for all $k \in K$), and $\mathcal{E}_f$ includes all events that are not elements of $\mathcal{E}_{h,1}$, $\mathcal{E}_{h,2}$, ..., or $\mathcal{E}_{h,K}$. Each classifier in the system seeks different types of hostile targets (e.g., one is trained to detect trucks, another is trained to detect artillery, etc.) The decisions from each classifier are sent to a fusion center or combiner, where a fusion rule is applied to the labels. The result is the decision for the classifier system in terms of the system label set $\mathcal{L}_S = \{f_S, h_S\}$. Figure 6 shows a two-classifier system in which each classifier can output two labels for an arbitrary fusion rule $R(l_1, l_2)$.

3.4.1 Joint State Probabilities Matrix. For the example, each classifier has 4 possible output states, and the associated probabilities are defined in their JPMs. Thus, there are $4 \cdot 4 = 16$ possible combinations of output states for a two-classifier system in which each classifier outputs two labels. In general, there are

$$\prod_{k=1}^{K} 2 \cdot m_k = 2^K \prod_{k=1}^{K} m_k$$

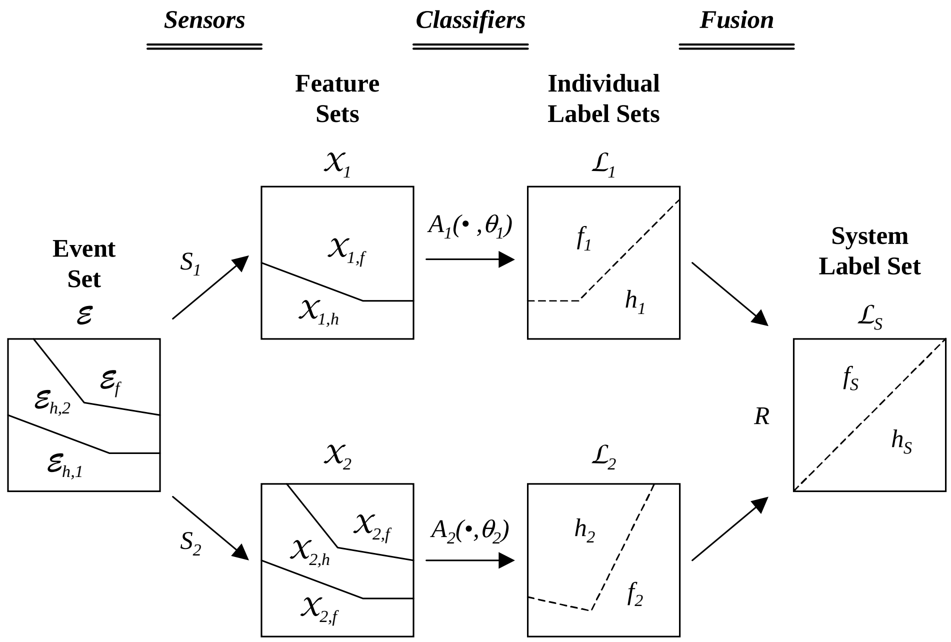

Figure 6: Event, Feature and Label Sets for a Two-Classifier System Combining Decisions *Across* Target Types

combinations of output states. Assuming statistical independence between the individual classifiers, one could compute the probability of a particular scenario by multiplying the probabilities of the associated output states. One way of accomplishing this is to enumerate all combinations of individual output states to determine the probability of the system being in a particular state, but a mathematical mechanism for determining the state combinations might enable the application of analysis techniques to help evaluate the performance of various fusion rules.

The Kronecker product is an operation that makes this possible. The Kronecker product, or tensor product, of two matrices multiplies each element of one matrix by each element of the other matrix in the following manner [11]. Assume A and B are 2×2 matrices. The Kronecker product

of A and B is

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{bmatrix}.$$

Note that $A \otimes B$ consists of all possible products of an A-matrix entry with B-matrix entry. Some fundamental properties of the Kronecker product are given in [4] and [11], and Van Loan notes the widening use of the operation and lists some areas where Kronecker product research is thriving: signal processing, image processing, semidefinite programming, quantum programming, and fast Fourier transforms [27]. The Kronecker product is defined for any pair of matrices of any dimensions, but for this application we will only be working with the $m_k \times 2$ CPMs and JPMs.

Given the JPMs for any two classifiers (1 and 2) seeking different types of targets, the Kronecker product of the two gives an $m_1 m_2 \times 4$ matrix in which each element contains the probability of a particular output state combination. This matrix is called the Joint State Probabilities Matrix (JSPM). If we denote the complement of x as $\overline{x} = (1 - x)$, the JSPM for two 2×2 JPMs is below.

$$J_1 \otimes J_2 = \begin{bmatrix} (1-\alpha_1)(1-P_{FA,1}) & \alpha_1(1-P_{D,1}) \\ (1-\alpha_1)P_{FA,1} & \alpha_1 P_{D,1} \end{bmatrix} \otimes \begin{bmatrix} (1-\alpha_2)(1-P_{FA,2}) & \alpha_2(1-P_{D,2}) \\ (1-\alpha_2)P_{FA,2} & \alpha_2 P_{D,2} \end{bmatrix}$$
$$= \begin{bmatrix} \overline{\alpha}_1\overline{P}_{FA,1}\overline{\alpha}_2\overline{P}_{FA,2} & \overline{\alpha}_1\overline{P}_{FA,1}\alpha_2\overline{P}_{D,2} & \alpha_1\overline{P}_{D,1}\overline{\alpha}_2\overline{P}_{FA,2} & \alpha_1\overline{P}_{D,1}\alpha_2\overline{P}_{D,2} \\ \overline{\alpha}_1\overline{P}_{FA,1}\overline{\alpha}_2 P_{FA,2} & \overline{\alpha}_1\overline{P}_{FA,1}\alpha_2 P_{D,2} & \alpha_1\overline{P}_{D,1}\overline{\alpha}_2 P_{FA,2} & \alpha_1\overline{P}_{D,1}\alpha_2 P_{D,2} \\ \overline{\alpha}_1 P_{FA,1}\overline{\alpha}_2\overline{P}_{FA,2} & \overline{\alpha}_1 P_{FA,1}\alpha_2\overline{P}_{D,2} & \alpha_1 P_{D,1}\overline{\alpha}_2\overline{P}_{FA,2} & \alpha_1 P_{D,1}\alpha_2\overline{P}_{D,2} \\ \overline{\alpha}_1 P_{FA,1}\overline{\alpha}_2 P_{FA,2} & \overline{\alpha}_1 P_{FA,1}\alpha_2 P_{D,2} & \alpha_1 P_{D,1}\overline{\alpha}_2 P_{FA,2} & \alpha_1 P_{D,1}\alpha_2 P_{D,2} \end{bmatrix}$$

As the rows and columns of the CPM and JPM correspond with labels and *truth*, respectively, the rows of the JSPM correspond with specific *pairs* of labels, and the columns correspond with

Table 7: Joint State Probabilities Matrix Composed from 2×2 JPMs.

		Feature Sets			
		$\mathcal{X}_{1,f} \times \mathcal{X}_{2,f}$	$\mathcal{X}_{1,f} \times \mathcal{X}_{2,h}$	$\mathcal{X}_{1,h} \times \mathcal{X}_{2,f}$	$\mathcal{X}_{1,h} \times \mathcal{X}_{2,h}$
Label Sets	f_1, f_2	$\overline{\alpha_1}\overline{P}_{FA,1}\overline{\alpha}_2\overline{P}_{FA,2}$	$\overline{\alpha_1}\overline{P}_{FA,1}\alpha_2\overline{P}_{D,2}$	$\alpha_1\overline{P}_{D,1}\overline{\alpha}_2\overline{P}_{FA,2}$	$\alpha_1\overline{P}_{D,1}\alpha_2\overline{P}_{D,2}$
	f_1, h_2	$\overline{\alpha_1}\overline{P}_{FA,1}\overline{\alpha}_2 P_{FA,2}$	$\overline{\alpha_1}\overline{P}_{FA,1}\alpha_2 P_{D,2}$	$\alpha_1\overline{P}_{D,1}\overline{\alpha}_2 P_{FA,2}$	$\alpha_1\overline{P}_{D,1}\alpha_2 P_{D,2}$
	h_1, f_2	$\overline{\alpha_1} P_{FA,1}\overline{\alpha}_2\overline{P}_{FA,2}$	$\overline{\alpha_1} P_{FA,1}\alpha_2\overline{P}_{D,2}$	$\alpha_1 P_{D,1}\overline{\alpha}_2\overline{P}_{FA,2}$	$\alpha_1 P_{D,1}\alpha_2\overline{P}_{D,2}$
	h_1, h_2	$\overline{\alpha_1} P_{FA,1}\overline{\alpha}_2 P_{FA,2}$	$\overline{\alpha_1} P_{FA,1}\alpha_2 P_{D,2}$	$\alpha_1 P_{D,1}\overline{\alpha}_2 P_{FA,2}$	$\alpha_1 P_{D,1}\alpha_2 P_{D,2}$

specific occurrences in *truth*, or Cartesian products of the different feature sets. Table 7 illustrates this point.

Definition III.4. *The* ***Joint State Probabilities Matrix (JSPM)****, denoted S_J, for a system of classifiers $\{1, 2, ..., K\}$ seeking K different targets is an $\prod_{k=1}^{K} m_k \times 2^K$ matrix in which the columns correspond with truth, the rows correspond with combinations of output labels from the individual classifiers, and cell (i, j) gives the probability of the classifier system outputting the combination of labels i when the true state of the system is j. The JSPM gives the probabilities of all possible states for the classifier system, and the sum of the elements of the JSPM is unity.*

Lemma III.1. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. Assuming statistical independence between the individual classifiers, the JSPM is formed by the Kronecker product of the JPMs for each classifier in the system.*

Proof III.1. *Let l_k be a generic output label for each classifier $k \in K$, and let $\mathcal{X}_{k,true}$ be a generic true state relative to the target sought by classifier k. Since the classifiers are statistically independent and each classifier seeks a different type of target, the probability for a given combination of labels and truth is*

$$
\begin{aligned}
\Pr\{l_1 \times l_2 \times \cdots \times l_K \cap \mathcal{X}_{1,true} \times \mathcal{X}_{2,true} \times \cdots \times \mathcal{X}_{K,true}\} &= \Pr\{l_1 \cap \mathcal{X}_{1,true} \times \cdots \times l_K \cap \mathcal{X}_{K,true}\} \\
&= \Pr\{l_1 \cap \mathcal{X}_{1,true}\} \cdot ... \cdot \Pr\{l_K \cap \mathcal{X}_{K,true}\} \\
&= \prod_{k=1}^{K} \Pr\{l_k \cap \mathcal{X}_{k,true}\} \qquad (1)
\end{aligned}
$$

Note that each of the terms in the product defined in Equation 1 is an element of a different JSPM, J_k. Since the Kronecker product of several matrices consists of all possible products of the entries of the individual matrices, the product in 1 must be an element of the Kronecker product defined by $J_1 \otimes J_2 \otimes \cdots \otimes J_K$.

Furthermore, it can be shown that the sum of the elements of a matrix formed from the Kronecker product of several JPMs (or for that matter, any matrices whose elements sum to one) is unity. Given $C_{D\times E}$ and $F_{G\times H}$, the Kronecker product $C_{D\times E} \otimes F_{G\times H}$ is a $DG \times EH$ matrix whose elements represent each combination of the cells in C and F (i.e., $c_{de}f_{gh}$ for any allowable values of $d, e, g,$ and h). The sum of the elements can be written

$$\begin{aligned} \textit{Sum of the Elements in } C_{D\times E} \otimes F_{G\times H} &= \sum_{d=1}^{D}\sum_{e=1}^{E}\sum_{g=1}^{G}\sum_{h=1}^{H} c_{de}f_{gh} \\ &= \sum_{d=1}^{D}\sum_{e=1}^{E} c_{de} \sum_{g=1}^{G}\sum_{h=1}^{H} f_{gh} \end{aligned} \quad (2)$$

Recall that the sum of the elements in C and F is also unity. That is, $\sum_{d=1}^{D}\sum_{e=1}^{E} c_{de} = 1$ and $\sum_{g=1}^{G}\sum_{h=1}^{H} f_{gh} = 1$. Inserting these results into Equation 2 gives the simple result that $1(1) = 1$. Since the elements of $J_1 \otimes J_2 \otimes \cdots \otimes J_K$ correspond with the appropriate values in the JSPM 1, and since $J_1 \otimes J_2 \otimes \cdots \otimes J_K$ satisfies the properties of a JSPM 2, $J_1 \otimes J_2 \otimes \cdots \otimes J_K$ must be a JSPM.

3.4.2 Combined Prior Probabilities Matrix. Note that the JSPM is equal to $J_1 \otimes J_2 \otimes ... \otimes J_K = C_1\rho_1 \otimes C_2\rho_2 \otimes \cdots \otimes C_2\rho_2$. One property of the Kronecker product is that $AB \otimes CD = (A \otimes C)(B \otimes D)$ for any matrices A, B, C and D [11]. Thus, the JSPM can be decomposed into the matrix product of $C_1 \otimes C_2 \otimes \cdots \otimes C_K$ and $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_K$. We will call the matrix composed by $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_K$ the Combined Prior Probabilities Matrix (CPPM) and represent it with P.

A CPPM for a system in which two classifiers seek two types of targets is given by

$$P = \rho_1 \otimes \rho_2 = \begin{bmatrix} (1-\alpha_1)(1-\alpha_2) & 0 & 0 & 0 \\ 0 & (1-\alpha_1)\alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_1(1-\alpha_2) & 0 \\ 0 & 0 & 0 & \alpha_1\alpha_2 \end{bmatrix}.$$

The elements of P represent the *a priori* probabilities of each of the feature set combinations defined by the columns of the JSPM.

Definition III.5. *The **Combined Prior Probabilities Matrix (CPPM)** for a system of classifiers $\{1, 2, ..., K\}$ seeking K different targets is a $2^K \times 2^K$ diagonal matrix in which the diagonal elements (j, j) give the a priori probability of the true state combinations defined in the j^{th} column of the JSPM. The trace of a CPPM is unity.*

Lemma III.2. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. Assuming statistical independence between the individual classifiers, the CPPM is formed by the Kronecker product of the PPMs for each classifier in the system.*

Proof III.2. *The proof is similar to the proof of Lemma III.1.*

3.4.3 Conditional State Probabilities Matrix. Recall that the JSPM for an MCS in which each classifier seeks a different type of target is given by

$$\begin{aligned} J_S &= J_1 \otimes J_2 \otimes ... \otimes J_K \\ &= C_1\rho_1 \otimes C_2\rho_2 \otimes ... \otimes C_2\rho_2 \\ &= (C_1 \otimes C_2 \otimes ... \otimes C_k)(\rho_1 \otimes \rho_2 \otimes ... \otimes \rho_K). \end{aligned} \tag{3}$$

The matrix, $C_1 \otimes C_2 \otimes ... \otimes C_k$, is called the Conditional State Probabilities Matrix (CSPM) and will be represented with S_C. The JSPM ($S_C P$) will henceforth be denoted S_J. The relationship

Table 8: Conditional State Probabilities Matrix Composed from 2×2 CPMs.

		Feature Sets			
		$\mathcal{X}_{1,f} \times \mathcal{X}_{2,f}$	$\mathcal{X}_{1,f} \times \mathcal{X}_{2,h}$	$\mathcal{X}_{1,h} \times \mathcal{X}_{2,f}$	$\mathcal{X}_{1,h} \times \mathcal{X}_{2,h}$
Label Sets	f_1, f_2	$\overline{P}_{FA,1}\overline{P}_{FA,2}$	$\overline{P}_{FA,1}\overline{P}_{D,2}$	$\overline{P}_{D,1}\overline{P}_{FA,2}$	$\overline{P}_{D,1}\overline{P}_{D,2}$
	f_1, h_2	$\overline{P}_{FA,1}P_{FA,2}$	$\overline{P}_{FA,1}P_{D,2}$	$\overline{P}_{D,1}P_{FA,2}$	$\overline{P}_{D,1}P_{D,2}$
	h_1, f_2	$P_{FA,1}\overline{P}_{FA,2}$	$P_{FA,1}\overline{P}_{D,2}$	$P_{D,1}\overline{P}_{FA,2}$	$P_{D,1}\overline{P}_{D,2}$
	h_1, h_2	$P_{FA,1}P_{FA,2}$	$P_{FA,1}P_{D,2}$	$P_{D,1}P_{FA,2}$	$P_{D,1}P_{D,2}$

between the CSPM and the CPM is analagous to the relationship between the JSPM and the JPM. The columns of S_C correspond with combinations of feature sets, and the rows correspond with combinations of labels. Table 8 gives an example of a CSPM.

Definition III.6. *The **Conditional State Probabilities Matrix (CSPM)** for a system of classifiers $\{1, 2, ..., K\}$ seeking K different targets is an $\prod_{k=1}^{K} m_k \times 2^K$ matrix in which the columns correspond with truth, the rows correspond with combinations of output labels from the individual classifiers, and cell (i, j) gives the conditional probability of the classifier system outputting the combination of labels i when the true state of the system is j. The sum of the elements in each column of the CSPM is unity.*

Lemma III.3. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. Assuming statistical independence between the individual classifiers, the CSPM is formed by the Kronecker product of the CPMs for each classifier in the system.*

Proof III.3. *The proof is similar to the proof of Lemma III.1. However, one must show that the Kronecker product of several stochastic matrices is also stochastic. This is a known result for the Kronecker product [27].*

3.4.4 Truth Matrix. Recall the 4×4 matrices presented in Tables 7 and 8, and note that the rightmost three columns correspond with the presence of at least one type of target. Further recall that the goal of the system is to determine if *any* targets are present. In the instance of any of the state combinations on these columns, at least one type of target is present. Since the states in the JSPM are mutually exclusive, the probability of either of two states occurring

Table 9: Result After Post-Multiplying a JSPM by a Truth Matrix.

		Truth	
		No Target	**Target Present**
Label Sets	f_1, f_2	$S_{J,(1,1)}$	$S_{J,(1,2)} + S_{J,(1,3)} + S_{J,(1,4)}$
	f_1, h_2	$S_{J,(2,1)}$	$S_{J,(2,2)} + S_{J,(2,3)} + S_{J,(2,4)}$
	h_1, f_2	$S_{J,(3,1)}$	$S_{J,(3,2)} + S_{J,(3,3)} + S_{J,(3,4)}$
	h_1, h_2	$S_{J,(4,1)}$	$S_{J,(4,2)} + S_{J,(4,3)} + S_{J,(4,4)}$

is the sum of the probabilities of the two states. Therefore, we can add the last three columns $(S_{J,(\cdot,2)} + S_{J,(\cdot,3)} + S_{J,(\cdot,4)})$ to arrive at the probability of a target being present under each possible label set. Conversely, the sum of the first column gives us the probability of no targets present. We can calculate both of these values by post-multiplying S_J by a truth matrix T, which takes the form

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Now we are left with a 4×2 matrix in which the rows correspond with each possible combination of labels, the first column corresponds with the absence of targets, and the second column corresponds with the presence of a target (or targets). Table 9 illustrates the resulting matrix for the 2×2 example. The cells of the matrix shown in Table 9 give the probabilities of a particular label when the system is in a particular state (target or not). For example, cell $(3, 2)$ gives $\Pr\{\text{Target Present} \cap (h_1, f_2)\}$.

Definition III.7. *A* ***Truth Matrix*** T *for an MCS combining decisions across target types is a* $2^K \times 2$ *matrix containing binary values in which row* i *corresponds with a column in the JSPM, the first column corresponds with the absence of hostile targets, and the second column corresponds with the presence of hostile targets. The columns of the Truth Matrix must be orthogonal. That is, if the* i^{th} *column of the JSPM corresponds with the presence of at least one target, the* $(i, 2)$ *cell of the Truth Matrix will contain a* 1*. Otherwise, the* $(i, 1)$ *cell of the Truth Matrix will contain a one.*

Lemma III.4. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. The truth matrix for such a system always has a one in the* $(1,1)$ *cell and zeros in all other cells of the first column. Consequently, the truth matrix also always has a zero in the* $(1,2)$ *cell and ones in the remaining cells of the second column.*

Proof III.4. *Assuming the CPMs for the individual classifiers were built with the previously defined convention (i.e., the first column corresponds with the absence of a target, the second column corresponds with the presence of a target, and the last row corresponds with the hostile label), the only column of the JSPM corresponding with a complete absence of targets must be the first column.*

3.4.5 Fusion Rule Matrix. Recall that a logical fusion rule selects combinations of labels for which the system concludes that a target is present. For example, the AND fusion rule will conclude that a target is present only if all classifiers in the ensemble conclude that a target is present. The AND rule corresponds with only the last row of the matrix in Table 9 (h_1, h_2). Since the AND rule will lead the system to conclude a target is present if and only if all classifiers determine a target is present, AND is a generally a conservative rule, which makes a false alarm less likely but also gives a lower probability of detection. A visual depiction of the results of the AND rule for the previous example is given in Figure 7.

The OR rule will decide that a target is present if any of the classifiers identify a target. The OR rule corresponds with rows two through four of the matrix in Table 9, or the intersection of the label sets $(f_1, h_2) \cup (h_1, f_2) \cup (h_1, h_2)$. The OR rule is more aggressive than the AND rule, usually yielding a higher false alarm rate as well as a higher detection probability. An illustration of the results of the OR rule for the previous example is given in Figure 8.

Once again we use the fact that the events defined by the cells of the matrix in Table 9 are mutually exclusive. If we want to determine the probability of identifying a target under a particular fusion rule we can add the appropriate cells from the second column If we want to determine the probability of identifying a target when none exists we can add the corresponding three cells from

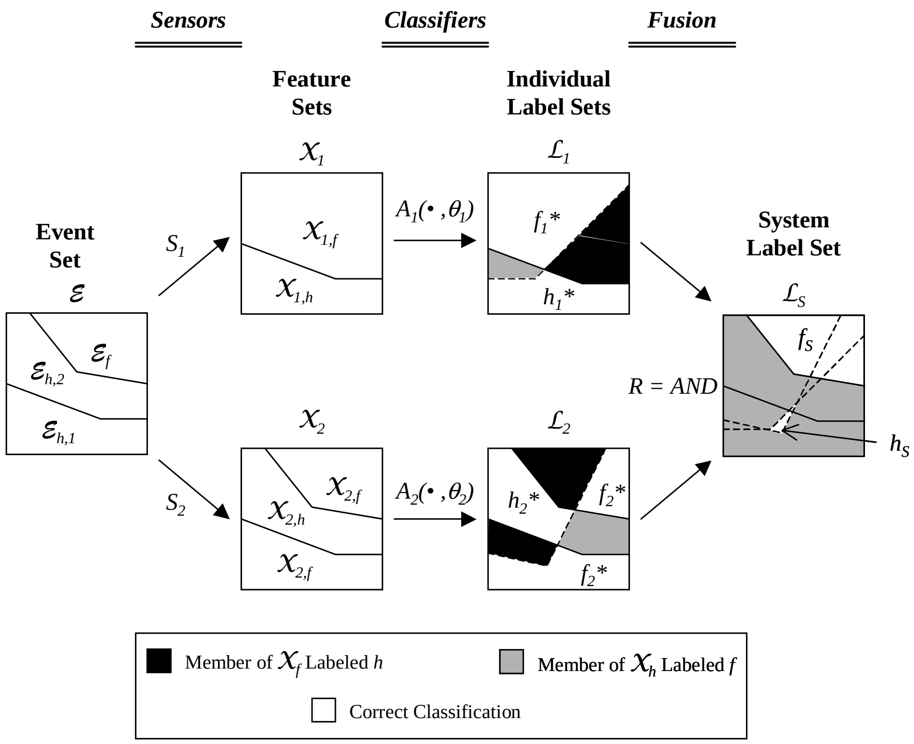

Figure 7: Classification Errors for a Two-Classifier System in Which Decisions are Combined Using the Logical AND Rule.

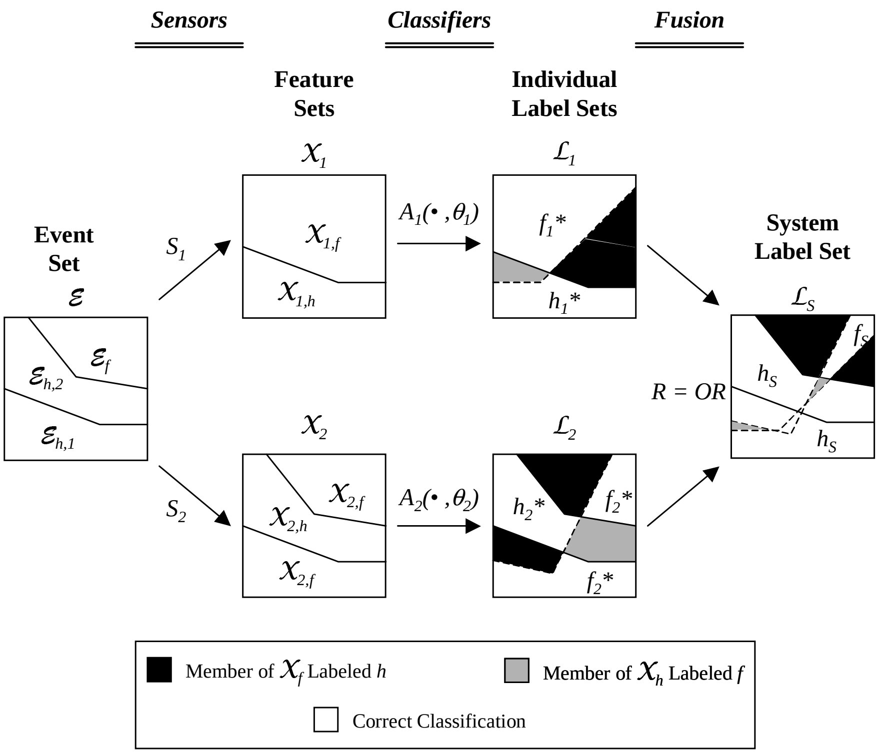

Figure 8: Classification Errors for a Two-Classifier System in Which Decisions are Combined Using the Logical OR Rule.

the first column. Recall that the $\vee$ rule corresponds with the last three rows of the matrix in Table 9. Now we can define a fusion vector as a column vector of zeros and ones, the ones corresponding to the rows appropriate for that rule. The fusion vector for the OR rule for our 2×2 example is

$$r_{OR} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

This fusion vector is similar to the "rule of engagement" defined in [22] and the rule vectors (for three classifiers) defined in [**?**].

Pre-multiplying the matrix in Table 9 (which was computed with the formula $S_C PT$) by the transpose of the fusion vector (to preserve dimensionality) gives us a vector containing the probabilities of correctly identifying a hostile target and misclassifying a friendly object, and pre-multiplying $S_C PT$ by the complementary vector to r_{OR}, $\overline{r}_{OR}^T = (1, 0, 0, 0)$, gives us the probabilities of misclassifying a hostile target and correctly identifying a friendly object.

$$r_{OR}^T S_C PT = \begin{bmatrix} \Pr\{h_S \cap \text{ No Target}\} & \Pr\{h_S \cap \text{ Target Present}\} \end{bmatrix}$$

and

$$\overline{r}_{OR}^T S_C PT = \begin{bmatrix} \Pr\{f_S \cap \text{ No Target}\} & \Pr\{f_S \cap \text{ Target Present}\} \end{bmatrix}$$

When the two vectors of are augmented in the form $[\overline{r}_{OR} \vdots r_{OR}]$, the result is the fusion rule matrix

$$R_{OR} = [\overline{r}_{OR} \vdots r_{OR}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

A more general definition is given by the following.

Definition III.8. *A* ***Fusion Rule Matrix*** *R is a $2^K \times m_S$ matrix containing binary values in which row i corresponds with a row in the JSPM, the first column corresponds with combinations of output labels for which the system concludes there is no target present, the last column corresponds with combinations of output labels for which the system concludes a hostile target is present, and the columns in between (if $m_S > 2$) correspond with intermediate fuzzy labels. For example, if the system is to conclude that a hostile target is present for a combination of output states corresponding with the i^{th} row of the JSPM, the (i, m_S) cell of the Fusion Rule Matrix will contain a 1. The columns of a Fusion Rule Matrix must be orthogonal.*

3.4.6 System Joint Performance Matrix.

Theorem III.1. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. If the classifiers are statistically independent, the system JPM, J_S, can be computed with the formula $R^T S_C P T$, where S_C and P are computed using Lemmas III.3 and III.2.*

Proof III.1. *Recall the proofs of Lemmas III.3 and III.2 and the result that $S_J = S_C P$. Then pre- and post-multiplying S_J by R^T and T, respectively, simply computes the sums of appropriate mutually exclusive probabilities. These sums correspond to the appropriate probabilities summarized in the JPM.*

- *The $(1, 1)$ cell is equivalent to $R_1^T S_J T_1$, which computes the sum of the cells of S_J that (a) are labeled friendly (by the rule matrix), and (b) are actually friendly (by the truth matrix).*

- *The* $(1,2)$ *cell is equivalent to* $R_1^T S_J T_2$*, which computes the sum of the cells of* S_J *that (a) are labeled friendly, and (b) are actually hostile.*
- *The* $(2,1)$ *cell is equivalent to* $R_2^T S_J T_1$*, which computes the sum of the cells of* S_J *that (a) are labeled hostile, and (b) are actually friendly.*
- *The* $(2,2)$ *cell is equivalent to* $R_2^T S_J T_2$*, which computes the sum of the cells of* S_J *that (a) are labeled hostile, and (b) are actually friendly.*

The result is a matrix with the following contstruction:

$$R^T S_C P T = \begin{bmatrix} \Pr\{A_S(x_S) = 1 \cap x_S \in \mathcal{X}_{S,f}\} & \Pr\{A_S(x_S) = 1 \cap x_S \in \mathcal{X}_{S,h}\} \\ \vdots & \vdots \\ \Pr\{A_S(x_S) = m_S \cap x_S \in \mathcal{X}_{S,f}\} & \Pr\{A_S(x_S) = m_S \cap x_S \in \mathcal{X}_{S,h}\} \end{bmatrix},$$

which satisfies the definition of a JPM.

3.4.7 System Prior Probabilities Matrix. Recall that the JPM for a classifier k can be computed with $J_k = C_k \rho_k$. Thus, one can post-multiply J_S by the inverse of the system prior probabilities matrix to compute C_S, but ρ_S has not yet been computed. Recall the matrix $P = \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_K$, or in terms of the example

$$P = \rho_1 \otimes \rho_2 = \begin{bmatrix} (1-\alpha_1)(1-\alpha_2) & 0 & 0 & 0 \\ 0 & (1-\alpha_1)\alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_1(1-\alpha_2) & 0 \\ 0 & 0 & 0 & \alpha_1\alpha_2 \end{bmatrix}.$$

Theorem III.2. *Consider an MCS in which each individual classifier is charged with identifying a different type of target. The system PPM,* ρ_S*, can be computed by pre-multiplying the CPPM by the transpose of the truth matrix* T^T *and post-multiplying by the truth matrix* T*.*

Proof III.2. *Recall that each row/column of the diagonal CPPM corresponds with a particular combination of true events (i.e., the Cartesian product of two particular feature sets) defined by the columns of the CSPM and JSPM. The last* $2^K - 1$ *rows/columns of the CPPM correspond with any event where a target is present, and the first row/column corresponds with instances where no target is present. Pre- and post-multiplying by the truth matrix* T *computes the sums of appropriate mutually exclusive probabilities. These sums correspond to the appropriate probabilities summarized in the PPM.*

- *The* $(1, 1)$ *cell is equivalent to* $T_1^T PT_1$*, which computes the sum of the cells of* P *that coincide with the absence of a target.*
- *The* $(1, 2)$ *and* $(2, 1)$ *cells are equivalent to* $T_1^T PT_2$ *and* $T_2^T PT_1$*, respectively. The result for either is always zero, since the columns of* T *are orthogonal.*
- *The* $(2, 2)$ *cell is equivalent to* $T_2^T PT_2$*, which computes the sum of the cells of* P *that coincide with the presence of at least one target.*

  The result is a matrix with the following construction,

$$T^T PT = \begin{bmatrix} \Pr\{\textit{No Target}\} & 0 \\ 0 & \Pr\{\textit{Target Present}\} \end{bmatrix},$$

which satisfies the definition of a PPM.

3.4.8 System Conditional Performance Matrix. Using the previously developed formula, one can now compute

$$C_S = J_S \rho_S^{-1} = R^T S_C P T \rho_S^{-1} = \begin{bmatrix} \Pr\{A_S(x_S) = 1 \mid x_S \in \mathcal{X}_{S,f}\} & \Pr\{A_S(x_S) = 1 \mid x_S \in \mathcal{X}_{S,h}\} \\ \vdots & \vdots \\ \Pr\{A_S(x_S) = m_S \mid x_S \in \mathcal{X}_{S,f}\} & \Pr\{A_S(x_S) = m_S \mid x_S \in \mathcal{X}_{S,h}\} \end{bmatrix}. \tag{4}$$

3.4.9 Summary. This section provided definitions for the CPM, JPM, PPM, JSPM, CPPM, CSPM, Truth Matrix, and Fusion Rule Matrix. Moreover, this section contained the derivation of a formula for computing the system PPM, CPM, and JPM for an MCS in which each individual classifier is charged with identifying a different type of target and the individual decisions are combined using Boolean fusion rules.

3.5 Within Fusion State Probabilities

Consider a system made up of K classifiers, each trying to detect the same type of target. The set of events $\mathcal{E}$ is partitioned into 2 subsets: $\mathcal{E}_h$ consists hostile targets and $\mathcal{E}_f$ includes all events that are not elements of $\mathcal{E}_h$. Assume that the CPM and JPM are available for each classifier, and note that the classifiers all share the same PPM since they all seek the same type of target. One would like to be able to compute the system CPM, PPM, and JPM as before, but this scenario possesses some properties that necessitate some changes in the computations of the state probabilities matrices (CSPM and JSPM) and the CPPM. The decisions from each classifier are sent to a fusion center or combiner, where a fusion rule is applied to the labels. The result is the decision for the classifier system in terms of the system label set $\mathcal{L}_S = \{1, 2, ..., m_S\}$. Figure 9 shows a two-classifier system in which each classifier can output two labels for an arbitrary fusion rule R.

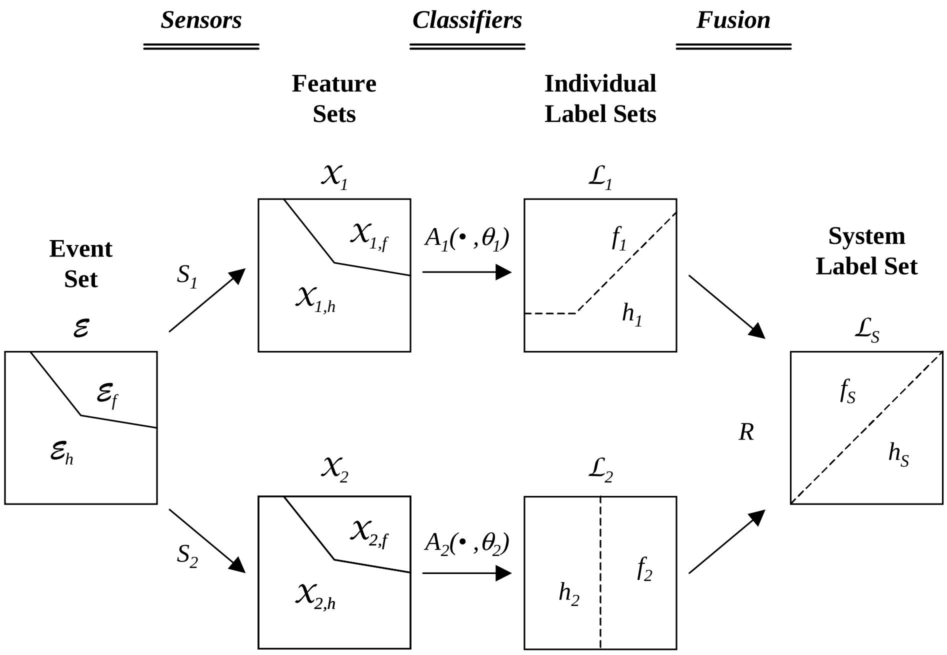

Figure 9: Event, Feature and Label Sets for a Multiple Classifier System Combining Decisions *Within* Target Types.

3.5.1 Conditional State Probabilities Matrix. Reconsider the CSPM for the system of classifiers fused *across* target types appearing in Table 8. This was the simplest example of a 4×4 CSPM, but it is adequate for illustrating the difference between the two cases. The second and third columns of the CSPM in Table 8 correspond with events where one type of target is present and the other type is not. A scenario such as this is impossible for the system presently being considered, because all the classifiers are seeking the same type of target (i.e., there are only two possible *true* states). One can use the Kronecker product to compute the possible combinations of the elements from the CPMs, but the result must be modified to account for these impossibilities. A simple way of removing them is to post-multiply the Kronecker product of the CPMs by a $2^K \times 2$ matrix with ones in the $(1, 1)$ and $(2^K, 2)$ cells. The result is a $2^K \times 2$ matrix consisting of the first

and last columns of the Kronecker product term. For example,

$$S_{C,Within} = (C_1 \otimes C_2) \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \left(\begin{bmatrix} 1 - P_{FA,1} & 1 - P_{D,1} \\ P_{FA,1} & P_{D,1} \end{bmatrix} \otimes \begin{bmatrix} 1 - P_{FA,2} & 1 - P_{D,2} \\ P_{FA,2} & P_{D,2} \end{bmatrix} \right) \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \overline{P}_{FA,1}\overline{P}_{FA,2} & \overline{P}_{FA,1}\overline{P}_{D,2} & \overline{P}_{D,1}\overline{P}_{FA,2} & \overline{P}_{D,1}\overline{P}_{D,2} \\ \overline{P}_{FA,1}P_{FA,2} & \overline{P}_{FA,1}P_{D,2} & \overline{P}_{D,1}P_{FA,2} & \overline{P}_{D,1}P_{D,2} \\ P_{FA,1}\overline{P}_{FA,2} & P_{FA,1}\overline{P}_{D,2} & P_{D,1}\overline{P}_{FA,2} & P_{D,1}\overline{P}_{D,2} \\ P_{FA,1}P_{FA,2} & P_{FA,1}P_{D,2} & P_{D,1}P_{FA,2} & P_{D,1}P_{D,2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \overline{P}_{FA,1}\overline{P}_{FA,2} & \overline{P}_{D,1}\overline{P}_{D,2} \\ \overline{P}_{FA,1}P_{FA,2} & \overline{P}_{D,1}P_{D,2} \\ P_{FA,1}\overline{P}_{FA,2} & P_{D,1}\overline{P}_{D,2} \\ P_{FA,1}P_{FA,2} & P_{D,1}P_{D,2} \end{bmatrix}$$

Definition III.9. *The* ***Conditional State Probabilities Matrix (CSPM)*** *for a system of classifiers* $\{1, 2, ..., K\}$ *seeking one type of target is a* $2^K \times 2$ *matrix in which the first column corresponds with instances where the target is absent, the second column corresponds with instances where the target is present, the rows correspond with combinations of output labels from the individual classifiers, and cell* (i, j) *gives the conditional probability of the classifier system outputting the combination of labels* i *when the true state of the system is* j *(i.e.* $\Pr\{A_S(x_S) = (l_1, l_2, ..., l_K) \mid x_S \in \mathcal{X}_{S,j}\}$*). The sum of the elements in each column of the CSPM is unity.*

Lemma III.5. *Consider an MCS in which each individual classifier is charged with identifying the same type of target. Assuming statistical independence between the individual classifiers, the CSPM is formed by post-multiplying the Kronecker product of the CPMs for each classifier in the system by a* $2^K \times 2$ *matrix with ones in the* $(1,1)$ *and* $(2^K, 2)$ *cells.*

Proof III.5. *Let* l_k *be a generic output label for each classifier* $k \in K$*, and let* $\mathcal{X}_{true}$ *be a generic true state relative to the target sought. Since the classifiers are statistically independent, the probability for a given combination of labels and truth is*

$$
\begin{aligned}
\Pr\{l_1 \times l_2 \times ... \times l_K \mid x_1 \times x_2 \times ... \times x_K \in \mathcal{X}_{true}\} &= \frac{\Pr\{l_1 \times l_2 \times ... \times l_K \cap x_1 \times x_2 \times ... \times x_K \in \mathcal{X}_{true}\}}{\Pr\{x_1 \times x_2 \times ... \times x_K \in \mathcal{X}_{true}\}} \\
&= \frac{\Pr\{l_1 \cap x_1 \in \mathcal{X}_{true}\} \cdot ... \cdot \Pr\{l_K \cap x_K \in \mathcal{X}_{true}\}}{\Pr\{x_1 \in \mathcal{X}_{true}\} \cdot ... \cdot \Pr\{x_K \in \mathcal{X}_{true}\}} \\
&= \Pr\{l_1 \mid x_1 \in \mathcal{X}_{true}\} \cdot ... \cdot \Pr\{l_K \mid x_K \in \mathcal{X}_{true}\} \\
&= \prod_{k=1}^{K} \Pr\{l_k \mid x_k \in \mathcal{X}.\}. \qquad (5)
\end{aligned}
$$

Note that each of the terms in the product defined in Equation 5 is an element of a different CPM. Since the Kronecker product of several CPMs consists of all possible products of the entries of the individual CPMs, the product in Equation 5 must be an element of the Kronecker product defined by $C_1 \otimes C_2 \otimes ... \otimes C_K$*. Post-multiplying the result by a* $2^K \times 2$ *matrix with ones in the* $(1,1)$ *and* $(2^K, 2)$ *cells leaves only the first and last columns, the columns corresponding with the target being absent or present, respectively.*

3.5.2 Combined Prior Probabilities Matrix. In a *within* fusion system, computation of the CPPM is trivial. Note that the CSPM is $2^K \times 2$. Therefore, one needs to post-multiply the CSPM by a 2×2 CPPM to arrive at a properly dimensioned JSPM. Since each classifier seeks the same type of target the *a priori* probability of the system being in the *hostile* state is the same as the *a priori* probabilities for all the classifiers in the system. Thus, the 2×2 matrix required is simply the PPM shared by the individual classifiers in the system (i.e., $P = \rho$).

Definition III.10. *The* ***Combined Prior Probabilities Matrix (CPPM)*** *for a system of classifiers* $(1, 2, ..., K)$ *seeking one type of target is equivalent to the PPM for each classifier in the system.*

3.5.3 Joint State Probabilities Matrix. The JSPM S_J can now be computed with

$$S_J = S_C P.$$

Definition III.11. *The* ***Joint State Probabilities Matrix (JSPM)*** *for a system of classifiers* $\{1, 2, ..., K\}$ *seeking one type of target is a* $2^K \times 2$ *matrix in which the first column corresponds with instances when the target is absent, the second column corresponds with instances when the target is present, the rows correspond with combinations of output labels from the individual classifiers, and cell* (i, j) *gives the probability of the classifier system outputting the combination of labels* i *when the true state of the system is* j *(i.e.* $\Pr\{A_S(x_S) = (l_1, l_2, ..., l_K) \cap x_S \in \mathcal{X}_{S,j}\}$*). The sum of the elements in the JSPM is unity.*

Lemma III.6. *Consider an MCS in which each individual classifier is charged with identifying the same type of target. Assuming statistical independence between the individual classifiers, the JSPM is formed by post-multiplying the CPPM,* S_C*, by the CPPM,* P*.*

Proof III.6. *Using the definition of conditional probability, one can post-multiply a matrix with the form*

$$\begin{bmatrix} \Pr\{\textit{Label Combination } 1 \mid \textit{No Target}\} & \Pr\{\textit{Label Combination } 1 \mid \textit{Target}\} \\ \Pr\{\textit{Label Combination } 2 \mid \textit{No Target}\} & \Pr\{\textit{Label Combination } 2 \mid \textit{Target}\} \\ \vdots & \vdots \\ \Pr\{\textit{Label Combination } 2^K \mid \textit{No Target}\} & \Pr\{\textit{Label Combination } 2^K \mid \textit{Target}\} \end{bmatrix}$$

by a matrix with the form

$$\begin{bmatrix} \Pr\{\textit{No Target}\} & 0 \\ 0 & \Pr\{\textit{Target}\} \end{bmatrix}$$

to yield a matrix with the form

$$\begin{bmatrix} \Pr\{\textit{Label Combination } 1 \cap \textit{No Target}\} & \Pr\{\textit{Label Combination } 1 \cap \textit{Target}\} \\ \Pr\{\textit{Label Combination } 2 \cap \textit{No Target}\} & \Pr\{\textit{Label Combination } 2 \cap \textit{Target}\} \\ \vdots & \vdots \\ \Pr\{\textit{Label Combination } 2^K \cap \textit{No Target}\} & \Pr\{\textit{Label Combination } 2^K \cap \textit{Target}\} \end{bmatrix}.$$

3.5.4 Truth and Fusion Rule Matrices. Using the definitions above for a *within* system, the JSPM is a $2^K \times 2$ matrix in which each column corresponds with *truth*. Thus, the truth matrix is no longer necessary because of the structure of this special case. One could consider the $2^K \times 2$ matrix used to compute the CSPM (i.e., the matrix with ones in the $(1,1)$ and $(2^K, 2)$ cells) the truth matrix, because its purpose is to eliminate impossible scenarios leaving only the two possible states (friend or hostile). Fusion rule matrices are defined in exactly the same way as in an *across* fusion system, and the system PPM is equivalent to the PPM shared by the individual classifiers. The formulae for computing the CPM and JPM are identical:

$$J_S = R^T S_C P T \text{ and}$$

$$C_S = J_S \rho_S^{-1}.$$

3.5.5 Summary. This section provided definitions for the CPM, JPM, PPM, JSPM, CPPM, CSPM, Truth Matrix, and Fusion Rule Matrix for an MCS fusing decisions *within* target types. This section also reiterated the formulae for computing the system CPM and JPM for an MCS in which the individual decisions are combined using Boolean fusion rules.

3.6 Estimating ROC Curves

3.6.1 Overview. This section suggests methods for estimating the ROC curve using the system CPM for an MCS combining decisions *within* or *across* target types. Oxley and Bauer's method of ROC fusion can be applied for Logical AND and OR rules; however, other techniques are necessary for more complex rules (e.g., a majority vote). Liggins showed that there are 7 relevant rules other than the AND and OR for combining the decisions of three classifiers [18]. Moreover, as the number of classifiers in the system gets larger, there are even more relevant rules besides the AND and OR rules. One might hope to find a way to estimate the ROC curves for systems combined using more complex fusion rules. Of these, the majority vote seems to be the most complex.

3.6.2 ROC Fusion.

3.6.2.1 Logical AND. One can employ Oxley and Bauer's method to analytically estimate the system ROC curve for a *within* or *across* MCS of *any* size if the system only outputs two labels (friendly or hostile) [20]. The key to their formula for the AND rule was the observation that, under an AND rule, the system probability for assigning a hostile label is equal to the product of each of the individuals assigning a hostile label:

$$\Pr\{h_S\} = \Pr\{h_1\} \cdot \Pr\{h_2\}$$
$$((1-\alpha_S)\, P_{FA,S} + \alpha_S P_{D,S}) = ((1-\alpha_1)\, P_{FA,1} + \alpha_1 P_{D,1})\,((1-\alpha_2)\, P_{FA,2} + \alpha_2 P_{D,2})\,.$$

Maximizing both sides of the last equation with respect to the individual threshold values and manipulating the result allows one to derive a formula for the maximum $P_{D,S}$ for a given $P_{FA,S}$. The property can easily be adapted to account for any number of classifiers by using the property

$$\Pr\{h_S\} = \prod_{k=1}^{K} \Pr\{h_k\}.$$

3.6.2.2 Logical OR. Oxley and Bauer derived a similar formula for the OR rule. The key to this formula was the observation that, under an OR rule, the system probability for assigning a friendly label is equal to the product of each of the individuals assigning a friendly label:

$$\begin{aligned}
\Pr\{f_S\} &= \Pr\{f_1\} \cdot \Pr\{f_2\} \\
(1-\alpha_S)\,(1-P_{FA,S}) + \alpha_S\,(1-P_{D,S}) &= \left[(1-\alpha_1)\,(1-P_{FA,1}) + \alpha_1\,(1-P_{D,1})\right] \\
&\quad \cdot \left[(1-\alpha_2)\,(1-P_{FA,2}) + \alpha_2\,(1-P_{D,2})\right].
\end{aligned}$$

Minimizing both sides of the last equation with respect to the individual threshold values and manipulating the result gives a formula for $\min\left(-P_{D,S}\right)$ which is equivalent to $\max P_{D,S}$. This property can be adapted in a similar manner such that

$$\Pr\{f_S\} = \prod_{k=1}^{K} \Pr\{f_k\}$$

3.6.3 Lagrangian Optimization. Another method for estimating the system ROC curve is to use a Lagrangian formulation like the one used in CFAR applications [28]. A typical Lagrangian equation $L = f(x) - \lambda(g(x))$ is appropriate in the following form:

$$L = P_{D,S} - \lambda(P_{FA,S} - p),$$

where p is any allowable false positive rate. Differentiating with respect to the threshold values (or P_{FA}) and the Lagrange multiplier gives a system of nonlinear equations (when $K \geq 3$) that can be solved to determine the maximum $P_{D,S}$.

3.6.4 Brute Force. If no other option is available, one can enumerate a subset of possible threshold (or P_{FA}) combinations and use the frontier of the results. This method is hardly scientific, but the computational complexity is not such that the method is impractical. Depending upon the

complexity of the fusion rule this method may be more *practical* than the others A drawback, however, is that the upper bound of the ROC is not guaranteed.

3.7 Summary

This chapter provided the following definitions for MCSs in (a) each classifier sought different types of targets and (b) each classifier sought the same type of target: Conditional Performance Matrix (CPM), Joint Performance Matrix (JPM), Prior Probabilities Matrix (PPM), Conditional State Probabilities Matrix (CSPM), Combined Prior Probabilities Matrix (CPPM), and Joint State Probabilities Matrix (JSPM). Also provided were derivations of formulae for the system JPM, PPM, and CPM for *within* and *across* fusion systems in which decisions are combined using Boolean fusion rules. Lastly, the reader is presented several methods for estimating the ROC curve for an MCS in which decisions are combined using Boolean fusion rules.

IV. Summary and Recommendations

4.1 Overview

This chapter summarizes the contributions of this thesis as applied to Multiple Classifier Systems in which decisions are combined using Boolean fusion rules. Additionally, the chapter will suggest areas of future research.

4.2 Summary of Contributions

The primary contribution of this thesis is a matrix algebraic formula for computing the Conditional and Joint Performance Matrices for a Boolean Multiple Classifier System. This thesis is the first known use of the Kronecker product for evaluating classifier system performance. Also presented were definitions and/or derivations for the following:

- Conditional Performance Matrix (CPM),
- Prior Probabilities Matrix (PPM),
- Joint Performance Matrix (JPM),
- Combined Prior Probabilities Matrix (CPPM),
- Conditional State Probabilities Matrix (CSPM),
- Joint State Probabilities Matrix (JSPM),
- Truth Matrix, and
- Fusion Rule Matrix.

Furthermore, several methods were presented for estimating an upper bound of the ROC curve for the MCS using the system CPM. The individual CPMs were used previously to determine optimal fusion rules [22]; however, that work did not take into account the possibility of varying the decision thresholds for the individual classifiers, nor did that work provide a methodology for

analyzing systems fusing *across* target types. Lastly, this thesis characterizes *Sensor Corroboration* rules that were considered by Liggins [18].

4.3 Recommendations for Future Research

The results of this research identify several potential areas for further research. First, the matrix algebraic formula for the system CPM suggests an underlying algebraic structure for Multiple Classifier Systems. Perhaps this structure can be extended to beyond Boolean MCSs to other types of MCSs (e.g., weighted voting systems).

Second, future analysts, engineers, and mathematicians may be able to exploit the algebraic structure in such a way as to improve classification accuracy. This might be accomplished by developing more clever ways to maximize detection probability for a given false alarm rate or through some other means entirely.

Third, this work took advantage of a commonly (ab)used assumption of statistical independence, even though statistical independence is not likely for MCSs in which the individuals seek similar target types. It may be possible to incorporate variance/covariance matrices to provide more realistic estimates of MCS performance.

Bibliography

1. Alsing, S. G. *The Evaluation of Competing Classifiers*. PhD dissertation, Air Force Institute of Technology, Wright Patterson AFB OH, March 2000.

2. Bauer, K. W., "Course Notes, OPER 685, Applied Multivariate Data Analysis." Graduate School of Engineering and Management, Air Force Institute of Technology, Wright Patterson AFB OH, Spring 2002.

3. Bradley, A. P. "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition*, *30*(7):1145–1159 (1997).

4. Brewer, J. W. "Kronecker Products and Matrix Calculus in System Theory," *IEEE Transactions on Circuits and Systems*, *25*(9):772–781 (September 1978).

5. Dasarathy, B. V. *Decision Fusion*. Los Alamitos, CA: IEEE Computer Society Press, 1994.

6. Egan, J. P. *Signal Detection Theory and ROC Analysis*. New York: Academic Press, 1975.

7. Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, *7*:179–188

8. Franke, J. and Mandler, E. "A Comparison of Two Approaches for Combining the Votes of Cooperating Classifiers." *11th International Conference on Pattern Recognition2*. 611–614. 1992.

9. Giacinto, G., Roli, F., and Bruzzone, L. "Combination of Neural and Statistical Algorithms for Supervised Classification of Remote-Sensing Images," *Pattern Recognition Letters*, *21*(5):385–397 (2000).

10. Giacinto, G., Roli, F., and Fumera, G., "Design of Effective Multiple Classifier Systems by Clustering of Classifiers."

11. Graham, A. *Kronecker Products and Matrix Calculus*. Chichester, UK: Ellis Horwood Limited, 1981.

12. Hanley, J. A. and McNeil, B. J. "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, *143*:29–36 (1982).

13. Hansen, L. K. and Salamon, P. "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10):993–1001 (October 1990).

14. Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3):226–239 (March 1998).

15. Klein, L. A. *Sensor and Data Fusion Concepts and Applications*. Bellingham Washington: SPIE - The International Society for Optical Engineering, 1993.

16. Kuncheva, L. I., WHITAKER, C. J., SHIPP, C. A., and DUIN, R. P. W. "Limits on the Majority Vote Accuracy in Classifier Fusion," *Pattern Analysis and Applications (to appear)*

17. Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. W. "Is independence good for combining classifiers?." *Proc. 15th International Conference on Pattern Recognition2*. 168–171. 2000.

18. Liggins, , IIM. E., Nebrich, M. A., Berlin, M. S., and Lazaroff, M., "Adaptive Multi-Image Decision Fusion Analysis."

19. Matan, O., "On Voting Enembles of Classifiers." AAAI-96 Workshop on Integrating Multiple Learned Models.

20. Oxley, M. E. and Bauer, K. W. *Classifier Fusion for Improved System Performance*. Technical Report, Air Force Institute of Technology, January 2002.

21. Perrone, M. P. and Cooper, L. N. *When Networks Disagree: Ensemble Methods for Hybrid Neural Networks*, 127–142. London, UK: Chapman Hall, 1993.

22. Ralston, J. M. *Bayesian Sensor Fusion for Minimum-Cost I.D. Declaration*. Technical Report IDA Paper P-3441, Institute for Defense Analyses, June 1999.

23. Roli, F., "Fusion of Multiple Classifiers." Course Notes, July 2002.

24. Saranli, A. and Demirekler, M. "On Output Independence and Complementariness in Rank-Based Multiple Classifier Decision Systems," *Pattern Recognition*, *34*:2319–2330 (2001).

25. Saranli, A. and Demirekler, M. "A Statistical Unified Framework for Rank-Based Multiple Classifier Decision Combination," *Pattern Recognition*, *34*:865–884 (2001).

26. Tumer, K. and Ghosh, J. "Analysis of Decision Boundaries in Linearly Combined Neural Classifiers," *Pattern Recognition*, *29*(2):341–348 (1996).

27. Van Loan, C. F. "The Ubiquitous Kronecker Product," *Journal of Computaional and Applied Mathematics*, *123*:85–100 (2000).

28. Varshney, P. K. *Distributed Detection and Data Fusion*. New York: Springer-Verlag, 1997.

29. Wackerly, D. D., Mendenhall III, W., and Scheaffer, R. L. *Mathematical Statistics with Applications*. Pacific Grove, CA: Duxbury, 2002.

30. Xu, L., Krzyzak, A., and Suen, C. Y. "Methods for Combining Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(3):418–435 (May/June 1992).

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
25-03-2003	Master's Thesis	Jun 2002 – Mar 2003

4. TITLE AND SUBTITLE

EVALUATING THE PERFORMANCE OF MULTIPLE CLASSIFIER SYSTEMS: A MATRIX ALGEBRA REPRESENTATION OF BOOLEAN FUSION RULES

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hill, Justin M., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENC/03M-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/SNAT
Attn: Devert Wicker
2214 Avionics Circle
WPAFB OH 45433-7321
DSN: 785-1115 x4250
e-mail: Devert.Wicker@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Given a finite collection of classifiers one might wish to combine, or fuse, the classifiers in hopes that the multiple classifier system (MCS) will perform better than the individuals. One method of fusing classifiers is to combine their final decision using Boolean rules (e.g., a logical OR, AND, or a majority vote of the classifiers in the system). An established method for evaluating a classifier is measuring some aspect of its Receiver Operating Characteristic (ROC) curve, which graphs the trade-off between the conditional probabilities of detection and false alarm. This work presents a unique method of estimating the performance of an MCS in which Boolean rules are used to combine individual decisions. The method requires performance data similar to the data available in the ROC curves for each of the individual classifiers, and the method can be used to estimate the ROC curve for the entire system. A consequence of this result is that one can save time and money by effectively evaluating the performance of an MCS without performing experiments.

15. SUBJECT TERMS

Sensor Fusion, Classification, Detection, Classifier System Performance, Kronecker Product, ROC Curve

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Mark E. Oxley, PhD (ENC)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	61	(937) 255-3636, ext 4515; e-mail: Mark.Oxley@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18